INCREMENTAL ADAPTIVE STRAREGIES FOR WIRELESS SENSOR NETWORKS

A thesis submitted in partial fulfilment of the requirement for the degree of

Master of Technology

In

Electronics and Communication Engineering

Specialization: Signal and Image Processing

By

Sowjanya Modalavalasa

Roll No: 213EC6272

Under the Guidance of

Dr. Ajit Kumar Sahoo

Department of Electronics and Communication Engineering

National Institute of Technology Rourkela

Rourkela, Odisha , 769008, India

May 2015

DEPARTMENT OF ELECRTONICS AND COMMUNICATION ENGINEERING,
NATIONAL INSTITUTE OF TECHNOLOGY,
ROURKELA, ODISHA -769008.

CERTIFICATE

This is to certify that the work done in the thesis entitled **Incremental Adaptive Strategies for wireless sensor Networks** by **Sowjanya Modalavalasa** is a record of an original research work carried out by him in National Institute of Technology, Rourkela under my supervision and guidance during 2014-2015 in partial fulfilment for the award of the degree in Master of Technology in Electronics and Communication Engineering (Signal and Image Processing), National Institute of Technology, Rourkela.

Place: NIT Rourkela

Date:

Dr. Ajit Kumar Sahoo

Asst. Professor

Dedicated to....

My Parents, my Brother and my Best Friend

DEPARTMENT OF ELECRTONICS AND COMMUNICATION ENGINEERING,
NATIONAL INSTITUTE OF TECHNOLOGY,
ROURKELA, ODISHA -769008.

DECLARATION

I certify that,

a. The work presented in this thesis is an original content of the research done by myself under the general supervision of my supervisor.

b. The project work or any part of it has not been submitted to any other institute for any degree or diploma.

c. I have followed the guidelines prescribed by the Institute in writing my thesis.

d. I have given due credit to the materials (data, theoretical analysis and text) used by me from other sources by citing them wherever I used them and given their details in the references.

e. I have given due credit to the sources (written material) used by quoting them where I used them and have cited those sources. Also their details are mentioned in the references.

Sowjanya Modalavalsa

ACKNOWLEDGEMENT

This research work is one of the significant achievements in my life and is made possible because of the unending encouragement and motivation given by so many people in every part of my life. It is an immense pleasure to have this opportunity to express my gratitude and regards to them.

First and foremost, I would like to express my profoundest gratitude and sincere thanks to **Dr. Ajit Kumar Sahoo**, Asst. Prof., in Department of Electronics and Communication Engineering**,** for his esteemed supervision, support and guidance during the tenure of my project work. His invaluable advices have motivated me a lot when I felt saturated in my work. His impartial feedback in every walk of the research has made me to approach a right way in excelling the work. I would also like to thank him for providing best facilities in the department.

I would like to express my gratitude and respect to Prof. K. K. Mahaptra, Prof. S. K. Meher, Prof. A.K. Sahoo, Prof. L. P. Roy, Prof. Samit Ari, Prof. S. Maity, Prof. A.K. Swain and Prof. D. P. Acharya for their guidance and suggestions throughout the M.Tech course. I take this opportunity to express gratitude to all of the Department faculty members for their help and support.

I would like to express my sincere thanks to the Ph.D. scholar Mr. Sanand Kumar for his valuable suggestions and support throughout my project work which inspired me a lot. I would like to express my heartfelt wishes to my brothers, friends and classmates whose company and support made me feel much better than what I am.

Lastly, I would like to express my love and heartful respect to my parents, and brother for their consistent support, encouragement in every walk of my life without whom I would be nothing.

Sowjanya Modalavalasa,

sowjanaya.mar15@gmail.com

ABSTRACT

Distributed wireless sensor networks play a key role due to its wide range of applications ranging from monitoring environmental parameters to satellite positioning. Adaptive algorithms are applied to the distributed networks to endow the network with adaptation capabilities. The distributed network consists of many small sensors deployed randomly in a geographic area, which are adaptive and share their local information. The efficiency of the adaptive distributed strategy relies on the mode of collaboration between the nodes and incremental mode of cooperation is considered throughout the work. A large number of adaptive algorithms are available in the literature, out of which choice is done according to the type of application, computational complexity and convergence rate. Least means square algorithm is the most popularly used adaptive algorithm due to its simplicity and least computational complexity. Distributed ILMS is used for parameter estimation and a spatial-temporal energy conservation relation is used to evaluate the steady state performance of the entire network. The simulated and theoretical steady state performances are compared.

Digital implementation of adaptive filters results in quantization errors and finite precision errors. ILMS suffers from drift problem, where the parameter estimate will go unbounded in non-ideal or practical implementations due to the continuous accumulation of quantization errors, finite precision errors and insufficient spectral excitation or ill conditioning of input sequence. They result in overflow and near singular auto correlation matrix, which provokes slow escape of parameter estimate to go unbound. The proposed method ILLMS uses the Leaky LMS algorithm, which introduces a leakage factor in the update equation, and so prevents the weights to go unbounded by leaking energy out. But the overall performance of ILLMS is similar to ILMS in terms of convergence speed and thus an incremental Modified Leaky LMS is proposed based on MLLMS algorithms which in turn derived from the LSE algorithm. LSE algorithm employs sum of exponentials of errors in its cost function and it results in convex and smooth error surface with more steepness, which results in faster convergence rate. ILLMS and IMLLMS algorithms are simulated and compared, where IMLLMS gives superior performance compared to ILLMS in terms of convergence rate and steady state values.

INDEX

LIST OF FIGURES

LIST OF ACRONYMS

LMS	:	Least Mean Square
NLMS	:	Normalized Least Mean Square
LLMS	:	Leaky Least Mean Square
ILMS	:	Incremental Least Mean Square
ILLMS	:	Incremental Leaky Least Mean Square
LSE	:	Least Sum of Exponentials
MLLMS	:	Modified Leaky Least Mean Square
IMLLMS	:	Incremental Modified Leaky Least Mean Square
MSE	:	Mean Square Error
EMSE	:	Excessive Mean Square Error
MSD	:	Mean Square Deviation
LMF	:	Least Mean Fourth
SOS	:	Second Order Statistics
HOS	:	Higher Order Statistics

CHAPTER 1

INTRODUCTION

The availability of embedded processors, low power micro sensors, actuators and radios empowered the utilization of distributed wireless sensor networks to an extensive variety of applications which include Precision agriculture, Environmental monitoring (air, water, soil, chemistry), Target localization, Habitat monitoring, Physiological monitoring, Transportation, Condition based maintenance, Military applications, Disaster relief management, Smart spaces, Factory instrumentation, Medical applications and Inventory tracking [1] [2] [3]. The traditional centralized processing includes micro sensors deployed in an area and are connected to the powerful central processor, where all the processing and estimation is performed. In contrast to the central processing, distributed wireless sensor networks are used in which signal processing is shared among the nodes. The need for wireless distributed sensing and processing is:

- If the exact position of the signal to be estimated is obscure in an area of observation, then distributed sensing permits us to deploy sensors in the vicinity of the phenomena to be observed than if only a single sensor is utilized. This results in better SNR and robustness towards environmental deterrents.
- A more advantageous approach for parameter estimation is wired networking of distributed sensors, but most of the environmental areas to be monitored will not have sufficient installed infrastructure for energy and communications. Thus the distributed micro sensors are forced to depend on limited local energy sources and communication channels.
- Even though the sensors are wired and distributed in the proximity of the phenomenon of interest, the centralized architecture for signal processing drains the energy and communication resources. Energy budget and communication constraints are the primary design criterion as the transmitted signal gets attenuated due to ground reflections arising from short antenna height. So the data should be processed as much as possible inside the nodes to reduce the bandwidth of the data to be transmitted.

The distributed sensor signal processing deals with collection and processing of local noisy observations of a parameter of interest in a geographical area where the micro sensors or nodes are deployed. All the nodes share their information according to the network topology and estimate the parameter of interest in a collaborative manner by utilizing their local noisy observations and the shared information from their immediate neighbours. In the traditional centralized processing, all the nodes will collect its noisy local data and send them to a centralized processor, which will perform the job of parameter estimation and broadcasts the result back to all the individual nodes. This involves a very powerful centralized processor and huge communication burden. Whereas in distributed adaptive solution all the nodes will have processing capabilities and perform the job of parameter estimation individually using their local data and information received form neighbour nodes. This saves a lot of energy and bandwidth.

The strategy of cooperation between the nodes will decide the data bandwidth and energy consumption. Basically there are three modes of collaboration namely incremental mode, diffusion mode, and probabilistic diffusion mode [4]. Each node in the adaptive distributed network is adaptive and the efficiency depends on the adaptation algorithm used and the mode of cooperation between the nodes. LMS algorithm is the most popularly used due to its less computational complexity and ease of implementation. Incremental mode of cooperation requires less power and communication and hence incremental mode is considered throughout the work.

The convergence speed of the LMS algorithm depends on the eigen value spread of the input fed to the nodes. Eigen value spread is defined as the ratio of largest eigen value to the smallest eigen value of the autocorrelation matrix of the input sequence. The largest eigen value limits the allowable range of step size for stability assurance and the smallest eigen value accounts for slow convergence rate. So the best convergence rate is achieved when all the eigen values are equal, which can be achieved by pre-whitening the data before processing. A rigorous mathematical analysis is done to observe and compare the simulated and theoretical results of the incremental LMS algorithm. LMS algorithm suffers from drift problem, where the parameter estimate will go unbounded even though the input sequence and error quantities are bounded. The accuracy and stability of LMS cannot be assured in non-ideal or practical scenarios where finite precision effects, quantization errors, inadequate input excitation comes into picture [7] [8].

A modification of the LMS algorithm is the LLMS algorithm primarily developed to overcome the drift problem [11]. LLMS employs a leakage factor in its weight update equation and so bounds the weights within limits by leaking some energy out. So ILLMS is developed to overcome the drift problem, accuracy and stability issues arising in ILMS. The performance of ILLMS is similar to ILMS both in convergence speed and MSE. A modified Leaky LMS (MLLMS) is the enhanced version of the LLMS is developed to obtain superior performance compared to ILLMS [19]. MLLMS is based on LSE algorithm [20], which is a generalization of the mixed norm gradient descent algorithms and employs sum of exponentials of errors in its cost function. So the cost function will have sum of even powers of error and so will have the combined effect of the second order statistic (SOS) algorithms like LMS, NLMS and under Higher order statistic (HOS) algorithms like LMS. This results in a convex and smooth error surface with more steepness assuring faster convergence rate and better MSE performance. Both LLMS and MLLMS are implemented in incremental case of the distributed network and are compared in terms of convergence speed and MSE.

1.1. Literature Survey

The practical applications of wireless sensor networks and the need for distributed sensor signal processing have been discussed in [1] [2] [3]. Various developments have been evolved for parameter estimation in an adaptive manner. The limitations of the centralized solution and the implementation of adaptive algorithms and specifically incremental gradient descent, incremental LMS algorithms are discussed in [4]. An incremental block LMS algorithm has been developed for distributed networks to reduce the computational complexity as in [5]. The adaptive algorithms have been implemented in different modes of cooperation as per the requirement as in [6]. The drift problem in LMS algorithm has been addressed in [7] [8] [9]. The solution to the drift problem has been analysed as in [10] [11] [12] [13]. Variants of the leaky LMS have been developed for better performance as in [14] [15] [16] [17] [19]. The performance and stability analysis of the LMF and like higher order statistic algorithms have been discussed in [18]. An MLLMS based on LSE, which is generalized mixed norm stochastic gradient algorithm has been developed for better convergence rate as in [19] [20].

1.2. Thesis Layout

Chapter 2 explains the basics of Incremental adaptive strategies for distributed networks and the practical applications of the distributed networks. The consensus strategy, steepest descent solution and adaptive incremental algorithms like incremental least mean squares (ILMS) are well discussed in this chapter. A rigorous mathematical analysis is done for properly understanding the performance analysis of the considered distributed network using spatio-temporal energy conservation relation. Steady sate analysis is done mathematically for the quantities of interest used to measure the efficiency of the algorithms like MSE, EMSE and MSD. Computer simulations are provided by comparing the simulated results and the theoretical results simulated for steady state behaviour at all the nodes in the distributed network. There is an excellent match between the simulated results and the theoretical values.

Chapter 3 deals with the problems arising in the ILMS and concludes with the solution for that. The drift problem arising due to practical implementations of Conventional LMS have been discussed. The reasons for the drift problem like finite precision effects, quantization errors and insufficient input excitation have been investigated theoretically and mathematically. The Leaky LMS has been quoted as the solution for the drift problem and mathematical analysis is done to show how LLMS overcomes the drift problem. Finally ILLMS is proposed to overcome drift problem arising in ILMS.

Chapter 4 describes the Incremental modified leaky LMS which outperforms ILLMS in terms of steady state behaviour and convergence rate. The modified leaky LMS is a modification of LLMS and is based LSE algorithm. The LSE, MLLMS are analysed in this chapter. IMLLMS is proposed to achieve better performance than ILLMS, since ILLMS performs similar to ILMS. So IMLLMS not only overcomes the drift problem, but also gives superior performance than ILLMS. Simulation results are provides comparing IMLLMS and ILLMS in a network of N nodes. The MSE, EMSE, MSD curves at individual nodes are compared for both the algorithms and the results are analysed.

Chapter 5 concludes the thesis and discusses regarding the future work and any further extensions to the present work.

CHAPTER 2

INCREMENTAL ADAPTIVE STRATEGIES OVER DISTRIBUTED NETWORK

2.1. Introduction

Distributed sensor networks comprises of many tiny sensors referred as nodes distributed in a geographic area over which we are interested to estimate the parameter of interest. Each sensor is referred as a node, which will collect the local noisy information about the parameter of interest available around it and processes the data and broadcast the information to the neighbours as per the network topology. All the nodes are adaptive and the network responds to real time excitations. The main objective is to estimate the parameter by processing and information sharing at each node such that the result is as accurate as the case if all nodes have information about the local data of all nodes in the network [1] [2]. The traditional solution for this linear estimation problem is that the local noisy observations sensed at each node will be transmitted to the centralized processor, which will estimate the parameter of interest and broadcast the parameter estimate back to all the nodes in the network. But this type of centralized processing requires a very powerful central processor and huge amount of communication between the processor and the nodes. Whereas, in distributed processing, each node will have their own tiny processor which will process the local noisy observations along with data received from their neighbourhood nodes. This significantly minimizes the amount of communication and processing.

2.2. Applications

The applications of distributed wireless sensor networks are enormous e.g. Environmental monitoring (air, water, chemistry), Habitat monitoring, Physiological monitoring, Transportation, disaster relief management, Precision agriculture, Target localization, Smart spaces, Military, Medical applications, Factory instrumentation and Inventory tracking [1] [2]. Some of the applications are discussed in detail to appreciate the importance of wireless sensor networks.

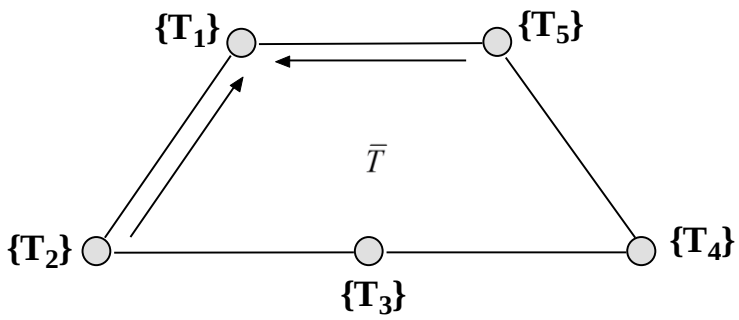

Fig. 2.1. Temperature Measurement in a distributed network with N nodes

Consider N nodes that are deployed in a geographical area for temperature measurement as shown in Fig.2.1. Each node in this distributed network will collect the local noisy observations T_i from the temperature sensors. The main aim is to equip each node with the knowledge about the average Temperature $\bar{T}$ of the geographic area considered. Consensus implementation is one of the distributed solutions provided for this problem, where each node estimates the parameter by combining the measurements received from its adjacent nodes that are connected to it. So the new measurement at each node is

$$x_1(i) \leftarrow \alpha_1 x_1(i-1) + \alpha_2 x_2(i-1) + \alpha_5 x_5(i-1)(1st\ node) \qquad (2.1)$$

Where $x_1(i)$ is the 1st node's updated measurement at ith iteration and α's are appropriately chosen coefficients. The same process is repeated where all the other nodes perform the same operation [1]. Proper choice of α and network topology results in convergence of all the nodes measurements to the average temperature $\bar{T}$.

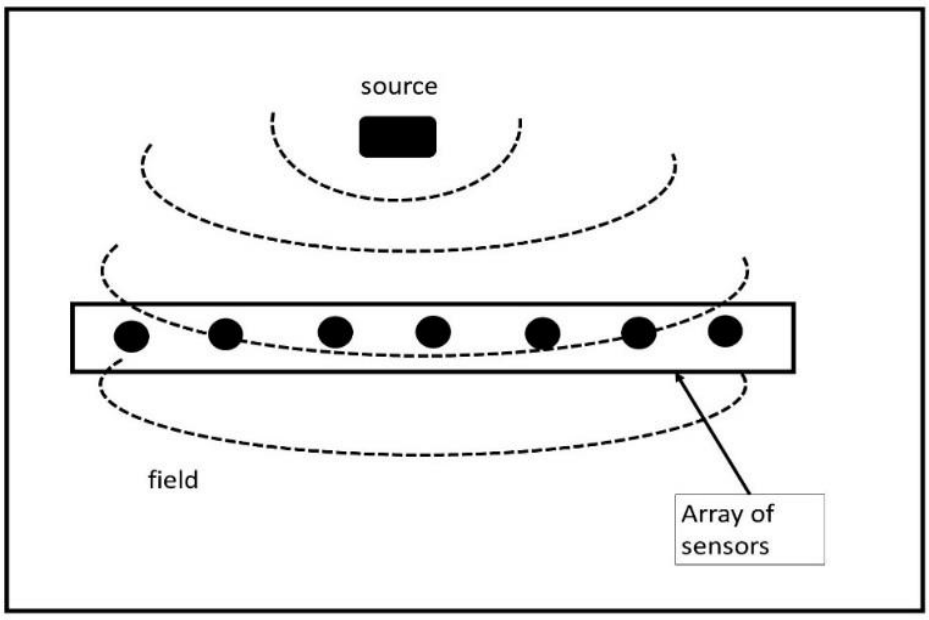

Fig. 2.1. Monitoring of a diffusion process by a network of sensors

An advanced application is to monitor chemical concentration in air, soil or water by collecting local information in time and space by a network of micro sensors as shown in Fig.2.2. The parameters $\{\theta_1, \theta_2, \theta_3\}$ dictating the diffusion of the chemical can be estimated from these local measurements by some diffusion equation subject to boundary conditions given as below:

$$\frac{\partial c(x,t)}{\partial t} = \theta_1 \frac{\partial^2 c(x,t)}{\partial x^2} + \theta_2 \frac{\partial c(x,t)}{\partial x} + \theta_3 c(x,t) + u(x,t) \tag{2.2}$$

Where $c(x,t)$ indicates the concentration at time t and at location x. Another application includes monitoring a moving target in a region, which is being monitored by a network of nodes or sensors [1] [3]. These sensors will communicate and share their noisy local data as per the network topology and will estimate the target location and trajectory.

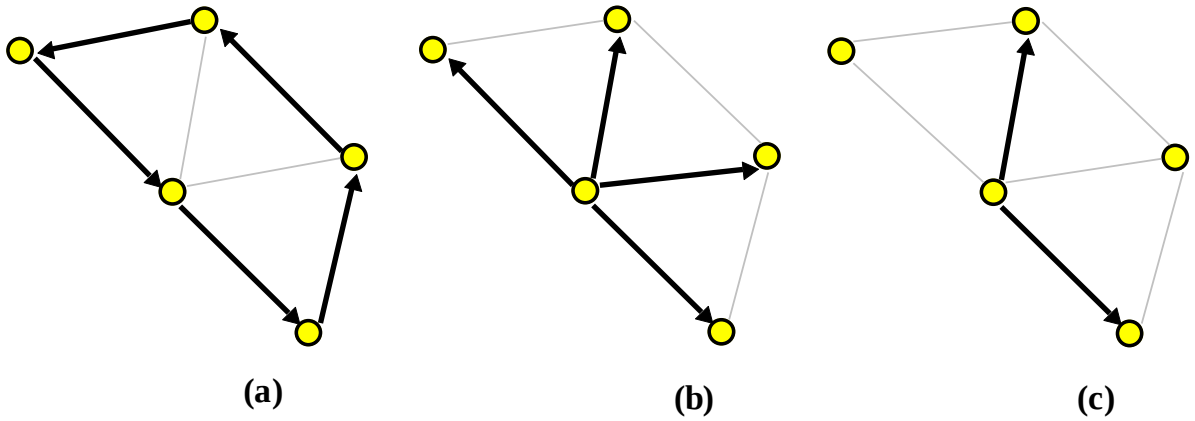

Fig. 2.2. Modes of collaboration (a) Incremental mode (b) Diffusion mode (c) Probabilistic diffusion mode

2.3. Modes of collaboration

The efficiency of any distributed sensor network implementation depends mostly on the amount of communication between the nodes i.e. mode of cooperation [4] [6]. Fig.2.3. depicts the three basic modes of collaboration available.

- In an Incremental mode of collaboration, information flows in a sequential or cyclic manner from one node to the adjoining node. This mode requires least amount of power and communication. This mode of cooperation may result in network failure if one of the nodes starts malfunctioning.

- In a Diffusion mode of cooperation, each and every node collaborates with all of its neighborhood nodes as determined by the network topology. Even though a node in the network fails, the effect is negligible as the nodes have more information from all their neighbor nodes and so the amount of communication, power and the computational complexity is higher than that of incremental mode of collaboration.
- In a Probabilistic diffusion mode of collaboration, each node is allowed to communicate with only a subset of neighbor nodes and the subset depends on the performance criteria and the requirement. So the communication burden is reduced compared to the Diffusion mode of cooperation, but still higher when compared with the Incremental mode of cooperation.

Incremental mode of cooperation is used throughout the work.

2.4. Consensus Strategy

Consensus implementation includes two steps. In first step each node collects its local noisy observations over a stipulated period of time and the parameter of interest will be estimated based on its individual data. During this first step there will be limited communication between the nodes, and in the next step all the nodes will merge their parameter estimates through several iterations to reach global estimate of the parameter of interest [3] [4].

Let us consider an example of a network of nodes to investigate the consensus strategy. Let us assume that each and every node has information regarding a data vector y_k and a data matrix H_k. The distorted and noisy measurement y_k is

$$y_k = H_k w^0 + v_k \tag{2.3}$$

Here w^0 is some unknown vector, which is to be estimated and v_k is some noise. Every node in the network evaluates the local cross correlation vector $\theta_k = {H_k}^* y_k$ and its autocorrelation matrix $R_k = {H_k}^* H_k$ for calculating the least-squares estimate of w^0. The local estimate of w^0 can be evaluated as $\widehat{w}_k = {R_k}^{-1}\theta_k$. Each node will estimate its local estimation $\widehat{w}_k$ in a similar way and then consensus iteration is applied to all the nodes to calculate $\hat{R}$ and $\hat{\theta}$ as below:

$$\hat{R} = \frac{1}{N}\sum_{k=1}^{N} R_k \quad \text{and} \quad \hat{\theta} = \frac{1}{N}\sum_{k=1}^{N} \theta_k \tag{2.4}$$

The global estimate of w^0 is given as $\widehat{w} = \hat{R}^{-1}\hat{\theta}$. The least squares solution in this style is an offline and non-recursive solution. If a node collects more entries than the other, then the problem arises for updating the optimal solution w^0 . The offline averaging limits the consensus solution in networks with limited communication resources and fast changing environments. To address these issues, distributed adaptive solution is to be developed so that the estimation will not require any data statistics or direct data i.e. model independent. The main objective is to develop distributed adaptive algorithms which endow the nodes with adaptation capabilities [4].

The main purpose of these distributed adaptive algorithms is:

- To develop an adaptive network structure with interconnected network of nodes which can react to the data in real time and track the changes in the statistical characteristics of the data as follows:
  - Every time a node picks up a new piece of data which is willingly utilised by the node for local parameter estimation.
  - Each node shares its local parameter estimates with the neighbour nodes based on the network topology.
- Distributed processing deals with "system of systems", which makes the task much more difficult as they process the data sequentially at all nodes, both in time and space. Each individual node will converge to different steady state MSE depending on the statistical properties of the data and background noise.

2.5. Estimation Problem & Adaptive Distributed Solution

Extensive works have been done on optimization problems of incremental distributed networks. In the distributed solution, using incremental strategies a cost function can be minimized by decoupling it into a sum of separate individual cost functions.

Let us consider a distributed network with N nodes as shown in Fig.2.4. Each and every node k has access to local noisy data realizations $\{d_k(i), u_{k,i}\}$ of the zero mean spatial data$\{d_k, u_k\}$,$k = 1,2,\cdots,N$, where d_k is the desired scalar and u_k is a regression input vector of size $1\times M$.

$$U \triangleq col\{u_1, u_2, \dots, u_N\}(N \times M)$$

$$d \triangleq col\{d_1, d_2, \dots, d_N\}(N \times 1)$$

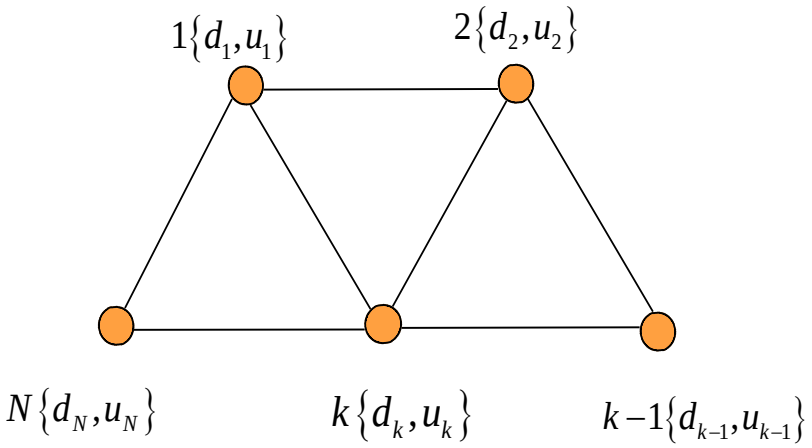

Fig.2.4. Distributed network with N nodes collecting local noisy information

The main intention is to estimate the vector w of size $M\times 1$ by using the above data collected from all N nodes and it should solve

$$\min_{w} \quad J(w)$$

Where $J(w)$ indicates the cost function which signifies the MSE, given as bellow:

$$J\,(w) = E\|d - Uw\|^2 \tag{2.5}$$

Where E is the expectation operator .The optimal solution w^0 can be obtained by using the orthogonality condition given as

$$E\|d - Uw\|^2 = 0$$

The solution to the above equation is given as

$$R_{du} = R_u w^0$$

Where $R_u = EU^*U \ \ (M \times M)$, $R_{du} = EU^*d = \sum_{k=1}^{N} R_{du,k}$

Computation of w^0 from above equations require knowledge of global statistical information$\{R_u, R_{du}\}$ at each node. An alternative solution for this problem is to compute the estimate centrally and broadcast the outcome back to the individual nodes, but this needs extensive communication and a powerful central processor. The sole solution is the

distributed solution, which allows limited communication between the nodes and equips the network with adaptation capabilities.

2.5.1. Steepest Descent Solution

A review of the steepest descent solution and its implementation is provided for better understanding of adaptive distributed solution. The cost function can be divided at each and every node as below:

$$\mathrm{J}\ (w) = \sum_{k=1}^{N} J_k(w)$$

Where $J_k(w)$ at each node is given as

$$J_k(w) \triangleq E|d_k - u_k w|^2$$

$$= {\sigma_{d,k}}^2 - R_{ud,k}w - w^* R_{du,k} + w^* R_{u,k} w \qquad (2.6)$$

And the second order moments are defined as

$${\sigma_{d,k}}^2 = E|d_k|^2,\ R_{u,k} = E{u_k}^* u_k, \text{ and } R_{du,k} = E d_k {u_k}^*$$

J (w) has been conveyed as the sum of N individual cost functions $J_k(w)$, one for each node k [4]. The traditional steepest descent solution for estimating w^0 given as below:

$$w_i = w_{i-1} - \mu[\nabla J(w_{i-1})]^*,\ \ w_{-1} = \text{initial condition}$$

$$= w_{i-1} - \mu \sum_{k=1}^{N} [\nabla J_k(w_{i-1})]^*$$

$$= w_{i-1} + \mu \sum_{k=1}^{N} \left(R_{du,k} - R_{u,k} w_{i-1}\right) \qquad (2.7)$$

Here $\mu > 0$ is the step size parameter, w_i is estimation of w^0 at i^{th} iteration and $\nabla J(w_{i-1})$ is the gradient of $J(w)$ which is calculated at w_{i-1} . For small step sizes, $w_i \to w^0$ as $i \to \infty$ for all the initial conditions. The same processes can be implemented in another way as follows.

The data processing in an adaptive distributed network is as shown in Fig.2.5. Let us define a cycle visiting each node, which will collect local noisy data and able to communicate with only it's immediate neighbour node. Let us assume that ${\psi_k}^{(i)}$ is the local estimate of w^0 at

time i and node k. so it has access to its neighbour's local estimate ${\psi_{k-1}}^{(i)}$. At every time instant i, start with the initial condition ${\psi_0}^{(i)} = w_{i-1}$ at first node and continue the process iteratively, which involves estimation of the parameter from local data , previous node's estimate and passing its new local estimate to its immediate neighbour node. At the ending of the process the local estimate at Nth node will give the global estimate i.e. ${\psi_N}^{(i)} = w_i$. This whole implementation can be written as below:

$$\begin{cases} {\psi_0}^{(i)} = w_{i-1} \\ {\psi_k}^{(i)} = {\psi_{k-1}}^{(i)} - \mu_k[\nabla J_k(w_{i-1})]^*, \quad k = 1,2,\cdots,N \\ w_i = {\psi_N}^{(i)} \end{cases} \tag{2.8}$$

For the above mentioned Steepest Descent Solution, the recursion for ${\psi_k}^{(i)}$ is over spatial index k.

2.5.2. Incremental Steepest Descent Solution

The solution provided in eq.2.8 is cooperative as each node uses information only from their immediate neighbour node, but requires each node to have global information w_{i-1} for calculation of $\nabla J_k(w_{i-1})$. To avoid this problem and to make the algorithm fully cooperative, incremental gradient algorithm is considered, where each node need ${\psi_{k-1}}^{(i)}$ from node k-1 to find the partial gradient $\nabla J_k\left({\psi_{k-1}}^{(i)}\right)$. So the Incremental Steepest Descent is as follows:

$$\begin{cases} {\psi_0}^{(i)} = w_{i-1} \\ {\psi_k}^{(i)} = {\psi_{k-1}}^{(i)} - \mu_k\left[\nabla J_k\left({\psi_{k-1}}^{(i)}\right)\right]^*, \quad k = 1,2,\cdots,N \\ w_i = {\psi_N}^{(i)} \end{cases} \tag{2.9}$$

The above solution is full pledged distributed solution as each node solely depends on its local data and communicates only with its immediate neighbor node. This minimizes energy consumption and communication burden.

2.5.3. Incremental Adaptive Solution

The adaptive solution provided in eq.2.9 requires information about the second order moments like cross correlation matrix ($R_{du,k}$) and autocorrelation matrix ($R_{u,k}$), which are required to calculate the local gradients vectors ∇J_k . An adaptive solution of eq.2.9 is

acquired by approximation the second order moments with their instantaneous values. It can be shown as below:

$$R_{du,k} \approx d_k(i)u_{k,i}^{*}, R_{u,k} \approx u_{k,i}^{*}u_{k,i} \tag{2.10}$$

So the algorithm uses the data $\{d_k(i), u_{k,i}\}$ at time i for adaptive implementation. This results in a distributed incremental adaptive solution or distributed ILMS algorithm as shown below [4]:

For each time $i \geq 0$, repeat:

K = 1,⋯ , N

$$\begin{cases} \psi_0^{(i)} = w_{i-1} \\ \psi_k^{(i)} = \psi_{k-1}^{(i)} + \mu_k u_{k,i}^{*}(d_k(i) - u_{k,i}\,\psi_{k-1}^{(i)}), \quad k = 1,2,\cdots,N \\ w_i = \psi_N^{(i)} \end{cases} \tag{2 11}$$

The operation of algorithm given in eq.2.11 is well explained in the Fig.2.5.

At every time i the node utilizes its local noisy data $\{d_k(i), u_{k,i}\}$ and the estimated weight vector $\psi_{k-1}^{(i)}$ received from it's the immediate neighbour node to accomplish the following three tasks:

- Calculate the local error value :$e_k(i) = d_k(i) - u_{k,i}\psi_{k-1}^{(i)}$;
- Update the weight estimate : $\psi_k^{(i)} = \psi_{k-1}^{(i)} + \mu_k u_{k,i}^{*} e_k(i)$;
- Pass the update weight estimate $\psi_k^{(i)}$ of node k to the immediate neighbour node k+1.

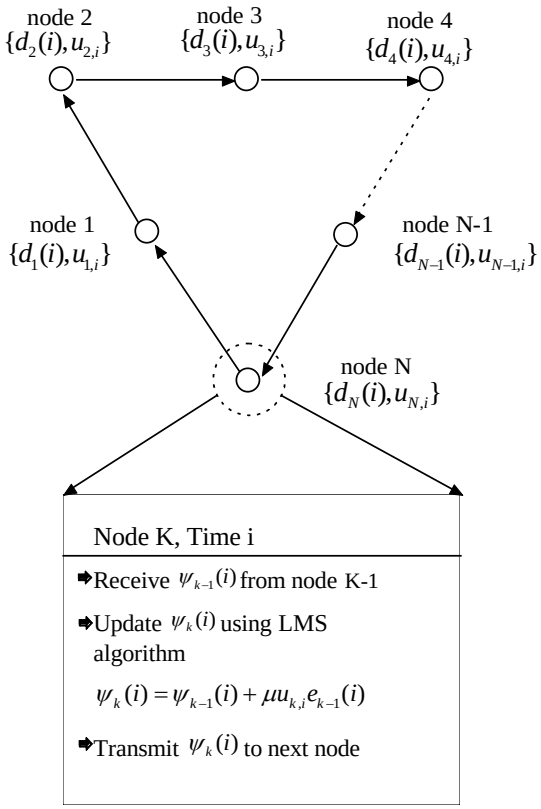

Fig.2.5. Data Flow and updation in Incremental LMS

2.6. Performance Analysis

The evaluation of the performance of the adaptive incremental solution involves determining how close the local estimate at each node (${\psi_k}^{(i)}$) reached the desired solution w^0. The difficulties which arise for investigating performance analysis of the network are challenging due to the following reasons [1] [4]:

- Each node k is distributed randomly in the geographical area and will be influenced by its local data statistics $\{R_{du,k}, R_{u,k}\}$ (Spatial data).
- Each node k distributed randomly in the geographical area is affected by its immediate neighbour through the incremental mode of collaboration (Spatial interaction).
- Each node is influenced by the local background noise with variance ${\sigma_{v,k}}^2$ (Spatial noise).

The parameters of interest for the performance analysis are Excessive Mean Square error (EMSE) and Mean Square Deviation (MSD), which will approach to zero asymptotically if the step size is decreased. The Mean Square error (MSE) will converge asymptotically to the

background noise in the network. The performance analysis is investigated relying on the energy conservation relation in both space and time since distributed adaptive algorithm involves both time (i) and space (k) indices. The energy flow across the interconnected nodes is to be studied since each node will stabilize at individual MSE in the steady state.

2.6.1. Data Model & Assumptions

The data model assumptions used for carrying out the performance analysis of the The desired unknown vector w^0 relates $\{d_k(i), u_{k,i}\}$ as

$$d_k(i) = u_{k,i}w^0 + v_k(i) \tag{2.12}$$

Where $v_k(i)$ is white noise sequence with variance ${\sigma_{v,k}}^2$ and independent of $\{d_l(j), u_{l,j}\}$ for all l,j.

- $u_{k,i}$ is independent of $u_{l,i}$ for k$\neq l$ (Spatial independence).
- $u_{k,i}$ is independent of $u_{k,j}$ for $i \neq j$ (time independence).

The model given above is considered from the literature of adaptive algorithms where all the nodes will try to estimate the unknown vectorw^0 and so referred as stationary model [4]. Distributed adaptive algorithms can also be used for non-stationary models.

2.6.2. Weighted Energy Conservation Relation

Let us assume the local error signals at each node k in the distributed network:

Weight error vector at time i : $\tilde{\psi}_k{}^{(i)} \triangleq w^0 - \psi_k{}^{(i)}$ (2.13)

A priori error : $e_{a,k}(i) \triangleq u_{k,i}\tilde{\psi}_{k-1}{}^{(i)}$ (2.14)

A posterior error : $e_{p,k}(i) \triangleq u_{k,i}\tilde{\psi}_k{}^{(i)}$ (2.15)

Output error : $e_k(i) \triangleq d_k(i) - u_{k,i}\psi_{k-1}{}^{(i)}$ (2.16)

The vector $\tilde{\psi}_k{}^{(i)}$ signifies the difference between the weight estimate at node k and the optimum weight vector w^0. The signal $e_k(i)$ signifies the estimation error, which is related to the a priori error by using the data model eq.2.12 is given as

$$e_k(i) = d_k(i) - u_{k,i}\psi_{k-1}{}^{(i)} = u_{k,i}w^0 + v_k(i) - u_{k,i}\psi_{k-1}{}^{(i)}$$

$$= e_{a,k}(i) + v_k(i) \tag{2.17}$$

$$\text{so here } E|e_k(i)|^2 = E\left|e_{a,k}(i)\right|^2 + {\sigma_{v,k}}^2 \tag{2.18}$$

The parameters of interest here are MSD, MSE and the EMSE, which can be obtained in steady state as follows:

$$\eta_k \triangleq E\left\|\tilde{\psi}_{k-1}^{\infty}\right\|^2 (MSD) \tag{2.19}$$

$$\varsigma_k \triangleq E\left|e_{a,k}(\infty)\right|^2 (EMSE) \tag{2.20}$$

$$\xi_k \triangleq E|e_k(\infty)|^2 = \varsigma_k + {\sigma_{v,k}}^2 (MSE) \tag{2.21}$$

The weight norm notation for a vector x and a hermitian positive definite matrix $\Sigma > 0$ is defined as

$$\|x\|_{\Sigma}^2 \triangleq x^{*}\Sigma x.$$

Then, according to our assumptions we have

$$\eta_k = E\left\|\tilde{\psi}_{k-1}^{(\infty)}\right\|_I^2 \quad \text{and} \quad \varsigma_k = E\left\|\tilde{\psi}_{k-1}^{(\infty)}\right\|_{R_{u,k}}^2 \tag{2.22}$$

So we need to evaluate means of two weighted norms. For that purpose let's first investigate the spatio-temporal energy conservation relation relating the local error variables. Let us define the weighted a priori and a posteriori local error signal at each node k as below:

$${e_{a,k}}^{\Sigma}(i) \triangleq u_{k,i}\Sigma\tilde{\psi}_{k-1}^{(i)} \; and \; {e_{p,k}}^{\Sigma}(i) \triangleq u_{k,i}\Sigma\tilde{\psi}_k^{(i)} \tag{2.23}$$

Where Σ is a Hermitian positive definite matrix, which we are free to choose. Using algorithm eq.2.11 and subtracting w^0 from both sides of the equation results as below:

$$\tilde{\psi}_k^{(i)} = \tilde{\psi}_{k-1}^{(i)} - \mu_k {u_{k,i}}^{*} e_k(i) \tag{2.24}$$

Multiplying eq.2.24 both side from left by $u_{k,i}\Sigma$ results in below equation

$$u_{k,i}\Sigma\ \tilde{\psi}_k^{(i)} = u_{k,i}\Sigma\ \tilde{\psi}_{k-1}^{(i)} - \mu_k \left\|u_{k,i}\right\|_{\Sigma}^2 e_k(i) \tag{2.25}$$

From eq.2.23 we get

$${e_{p,k}}^{\Sigma}(i) = {e_{a,k}}^{\Sigma}(i) - \mu_k \left\|u_{k,i}\right\|_{\Sigma}^2 e_k(i) \tag{2.26}$$

It follows,

$$e_k(i) = \frac{1}{\mu_k \|u_{k,i}\|_{\Sigma}^2}\left({e_{a,k}}^{\Sigma}(i) - {e_{p,k}}^{\Sigma}(i)\right) \tag{2.27}$$

Substituting eq.2.27 into eq.2.24 and rearranging terms, we get

$$\tilde{\psi}_k^{(i)} + \frac{{u_{k,i}}^{*}{e_{a,k}}^{\Sigma}(i)}{\|u_{k,i}\|_{\Sigma}^2} = \tilde{\psi}_{k-1}^{(i)} + \frac{{u_{k,i}}^{*}{e_{p,k}}^{\Sigma}(i)}{\|u_{k,i}\|_{\Sigma}^2} \tag{2.28}$$

Equalizing the weighted norm of both sides, and cancelling out the cross terms the result will contain the energy terms only, as shown below:

$$\left\|\tilde{\psi}_k^{(i)}\right\|_{\Sigma}^{2} + \frac{|e_{a,k}^{\Sigma}(i)|^2}{\|u_{k,i}\|_{\Sigma}^{2}} = \left\|\tilde{\psi}_{k-1}^{(i)}\right\|_{\Sigma}^{2} + \frac{|e_{p,k}^{\Sigma}(i)|^2}{\|u_{k,i}\|_{\Sigma}^{2}} \tag{2.29}$$

The above equation is the space-time weighted energy conservation relation, which shows the relation between the energies of several error variables in space and time. The above relation is for regular adaptive algorithms where no approximations are used.

2.6.3. Variance Relation

Subsequent analysis involves evaluation of performance at individual nodes based on the pace-time weighted energy conservation relation. The time index i is dropped for simplicity. Now by replacing eq.2.26 into eq.2.29 and rearranging terms we get

$$\left\|\tilde{\psi}_k^{(i)}\right\|_{\Sigma}^{2} = \left\|\tilde{\psi}_{k-1}^{(i)}\right\|_{\Sigma}^{2} - \mu_k e_{a,k}^{\Sigma *} e_k - \mu_k e_k^{*} e_{a,k}^{\Sigma} + \mu_k^{2} |u_k|_{\Sigma}^{2} |e_k|^2 \tag{2.30}$$

Using eq.2.30 and taking expectation of both sides

$$E\left\|\tilde{\psi}_k^{(i)}\right\|_{\Sigma}^{2} = E\left\|\tilde{\psi}_{k-1}^{(i)}\right\|_{\Sigma}^{2} - \mu_k E e_{a,k}^{\Sigma *} e_k - \mu_k e_k^{*} E e_{a,k}^{\Sigma} + \mu_k^{2} E |u_k|_{\Sigma}^{2} \left|e_{a,k}\right|^2 \tag{2.31}$$

Using eq.2.23 and weighted error definitions, we can expand the eq.2.31 in terms of regressor data and weighted error vector as follows:

$$E\left\|\tilde{\psi}_k^{(i)}\right\|_{\Sigma}^{2} =$$

$$E\left\|\tilde{\psi}_{k-1}^{(i)}\right\|_{\Sigma}^{2} - \mu_k E \tilde{\psi}_{k-1}^{*} \Sigma u_k^{*} u_k \tilde{\psi}_{k-1} - \mu_k E \tilde{\psi}_{k-1}^{*} u_k^{*} u_k \Sigma \tilde{\psi}_{k-1} +$$

$$\mu_k^{2} E \tilde{\psi}_{k-1}^{*} u_k^{*} u_k \Sigma u_k^{*} u_k \tilde{\psi}_{k-1} + \mu_k^{2} \sigma_{v,k}^{2} E \|u_k\|_{\Sigma}^{2} \tag{2.32}$$

Using the relation $\|x\|^2{}_A + \|x\|^2{}_B = \|x\|^2{}_{A+B}$, and by using this eq.2.32 can be rewritten as

$$E\left\|\tilde{\psi}_k^{(i)}\right\|_{\Sigma}^{2} = E\left(\left\|\tilde{\psi}_{k-1}^{(i)}\right\|_{\Sigma'}^{2}\right) + \mu_k^{2} \sigma_{v,k}^{2} E |u_k|_{\Sigma}^{2} \tag{2.33}$$

Where the term Σ' represents the stochastic weighted matrix given as

$$\Sigma' = \Sigma - \mu_k\left(u_k^{*} u_k \Sigma + \Sigma u_k^{*} u_k + \mu_k^{2} \|u_k\|_{\Sigma}^{2} u_k^{*} u_k\right) \tag{2.34}$$

Since u_k is the independence regressor data we can write as

$$E\left(\left\|\tilde{\psi}_{k-1}{}^{(i)}\right\|_{\Sigma'}{}^{2}\right) = E\left\|\tilde{\psi}_{k-1}{}^{(i)}\right\|_{E\Sigma'}{}^{2} \tag{2.35}$$

Again rewriting eq.2.33 and eq.2.34 as

$$E\left\|\tilde{\psi}_{k}{}^{(i)}\right\|_{\Sigma}{}^{2} = E\left(\left\|\tilde{\psi}_{k-1}{}^{(i)}\right\|_{\Sigma'}{}^{2}\right) + \mu_k{}^{2}\sigma_{v,k}{}^{2}E|u_k|_{\Sigma}{}^{2} \tag{2.36}$$

$$\Sigma' = \Sigma - \mu_k\left(u_k{}^{*}u_k\Sigma + \Sigma u_k{}^{*}u_k + \mu_k{}^{2}\|u_k\|_{\Sigma}{}^{2}u_k{}^{*}u_k\right) \tag{2.37}$$

So Σ' is now a deterministic matrix.

2.6.4. Gaussian Data

Equation 2.36 is the spatial variance equation and is used to perform the steady state analysis at every individual node k. From eq.2.37 one can conclude that Σ' totally depends on the regressor. So the further analysis of the performance of the network depends on the following three parameters:

$$R_{u,k} = Eu_k{}^{*}u_k, \quad \mathrm{E}\|u_k\|_{\Sigma}{}^{2} = Tr\left(R_{u,k}\Sigma\right), \text{ and } \mathrm{E}\|u_k\|_{\Sigma}{}^{2}u_k{}^{*}u_k \tag{2.38}$$

For simplicity during the calculation of $\mathrm{E}\|u_k\|_{\Sigma}{}^{2}u_k{}^{*}u_k$ the input vector is assumed to be Gaussian data. So let's assume that $\{u_k\}$ arises from circular Gaussian distribution. The Eigen value decomposition of auto correlation matrix is $R_{u,k} = U_k\Lambda_k U_k{}^{*}$, where $\Lambda_k$the diagonal matrix with Eigen value of is $R_{u,k}$ and U_k is the unitary matrix. Now the shifted quantities are

$$\bar{\psi}_k \triangleq U_k{}^{*}\tilde{\psi}_k,\ \bar{\psi}_{k-1} \triangleq U_k{}^{*}\tilde{\psi}_{k-1},\ \bar{u}_k \triangleq u_{kU_k},\ \bar{\Sigma} \triangleq U_k{}^{*}\Sigma U_k,\ \bar{\Sigma}' \triangleq U_k{}^{*}\Sigma' U_k \tag{2.39}$$

As the matrix U_k is unitary, $\left\|\tilde{\psi}_{k-1}\right\|_{\Sigma}{}^{2} = \left\|\bar{\psi}_{k-1}\right\|_{\bar{\Sigma}}{}^{2}\ and\ \|u_k\|_{\Sigma}{}^{2} = \|\bar{u}_k\|_{\bar{\Sigma}}{}^{2}$,by using these relation , eq.2.36 and eq.2.37 can be rewritten as

$$E\left\|\bar{\psi}_k\right\|_{\bar{\Sigma}}{}^{2} = E\left\|\bar{\psi}_{k-1}\right\|_{\bar{\Sigma}'}{}^{2} + \mu_k{}^{2}\sigma_{v,k}{}^{2}E\|\bar{u}_k\|_{\bar{\Sigma}}{}^{2} \tag{2.40}$$

$$\bar{\Sigma}' = \bar{\Sigma} - \mu_k E(\bar{u}_k{}^{*}\bar{u}_k\bar{\Sigma} + \bar{\Sigma}\bar{u}_k{}^{*}\bar{u}_k) + \mu_k{}^{2}E\|\bar{u}_k\|_{\bar{\Sigma}}{}^{2}\bar{u}_k{}^{*}\bar{u}_k \tag{2.41}$$

$$E\left\|\bar{\psi}_k\right\|_{\bar{\Sigma}}{}^{2} = Tr(\Lambda_k\bar{\Sigma}) \text{ and } E\bar{u}_k{}^{*}\bar{u}_k = \Lambda_k \tag{2.42}$$

$$E\|\bar{u}_k\|_{\bar{\Sigma}}{}^{2}\bar{u}_k{}^{*}\bar{u}_k = \Lambda_k Tr(\bar{\Sigma}\Lambda_k) + \gamma\Lambda_k\bar{\Sigma}\Lambda_k \tag{2.43}$$

Where, $\gamma = 1$ for circular complex data and $\gamma = 2$ for real data. Now substituting eq.2.42 and eq.2.43 into eq.2.40 and eq.2.41 we get

$$E\left\|\bar{\psi}_k\right\|_{\bar{\Sigma}}{}^{2} = E\left\|\bar{\psi}_{k-1}\right\|_{\bar{\Sigma}'}{}^{2} + \mu_k{}^{2}\sigma_{v,k}{}^{2}Tr(\Lambda_k\bar{\Sigma}) \tag{2.44}$$

$$\bar{\Sigma}' = \bar{\Sigma} - \mu_k(\Lambda_k\bar{\Sigma} + \bar{\Sigma}\Lambda_k) + {\mu_k}^2(\Lambda_k Tr(\bar{\Sigma}\Lambda_k) + \gamma\Lambda_k\bar{\Sigma}\Lambda_k) \tag{2.45}$$

2.6.5. Diagonalization

As we are free to choose Σ , let's choose Σ such that $\bar{\Sigma}' and\ \bar{\Sigma}$ will become diagonal which simplifies further analysis. Let us define the $M \times 1$ column vectors

$$\bar{\sigma} \triangleq diag\{\bar{\Sigma}\},\ \bar{\sigma}' \triangleq diag\{\bar{\Sigma}'\},\ \ \lambda_k \triangleq diag\{\Lambda_k\} \tag{2.46}$$

Where the $diag\{\ \}$ indication is used in two ways:

$\Lambda = diag\{\lambda\}$ represents diagonal matrix whose elements are the vector of λ .

$\lambda = diag\{\Lambda\}$ represents a vector containing main diagonal entries of Λ.

Using this concept eq.2.45 can be rewritten as

$$\bar{\sigma} = \left(I - 2\mu_k\Lambda_k + \gamma{\mu_k}^2{\Lambda_k}^2\right)\bar{\sigma} + {\mu_k}^2\left({\lambda_k}^T\bar{\sigma}\right)\lambda_k \qquad = \bar{F}_k\bar{\sigma} \tag{2.47}$$

Here the coefficient matrix $\bar{F}_k$ is defined by

$$\bar{F}_k \triangleq I - 2\mu_k\Lambda_k + \gamma{\mu_k}^2{\Lambda_k}^2 + {\mu_k}^2\lambda_k{\lambda_k}^T \tag{2.48}$$

So the expression eq.2.44 becomes

$$E\left\|\bar{\psi}_k\right\|_{diag\{\bar{\sigma}\}}^{\ 2} = E\left\|\bar{\psi}_{k-1}\right\|_{diag\{\bar{F}_k\bar{\sigma}\}}^{\ 2} + {\mu_k}^2{\sigma_{v,k}}^2\left({\lambda_k}^T\bar{\sigma}\right) \tag{2.49}$$

For simplicity $diag\{\ \}$ notation is dropped, now the equation becomes

$$E\left\|\bar{\psi}_k^{\ (i)}\right\|_{\bar{\sigma}_k}^{\ 2} = E\left\|\bar{\psi}_{k-1}^{\ (i)}\right\|_{\bar{F}_k\bar{\sigma}_k}^{\ 2} + {\mu_k}^2{\sigma_{v,k}}^2\left({\lambda_k}^T\bar{\sigma}_k\right) \tag{2.50}$$

The time index i is restored and $\left\{\bar{\sigma}, \bar{\sigma}^{'}\right\}$ are replaced by $\left\{\bar{\sigma}_k, \bar{\sigma}_k^{'}\right\}$ to show that weighting matrix could be bode dependent.

2.6.6. Steady State Behaviour

Let $\bar{p}_k \triangleq \bar{\psi}_k^{(\infty)}$ and $g_k \triangleq {\mu_k}^2 {\sigma_{v,k}}^2 {\lambda_k}^T$(a row vector). Then, for $i \to \infty$, the variance relation eq.2.50 can be rewritten as

$$E\|\bar{p}_k\|_{\bar{\sigma}_k}{}^2 = E\|\bar{p}_{k-1}\|_{\bar{F}_k\bar{\sigma}_k}{}^2 + g_k\bar{\sigma}_k, \;\; k = 1,2,\cdots,N \tag{2.51}$$

For evaluation of performance measurement we are after MSE, MSD, EMSE, which are defined as below:

$$\eta_k = E\|\bar{p}_{k-1}\|_q{}^2, \;\; q \triangleq diag\{I\} \quad (MSD) \tag{2.52}$$

$$\zeta_k = E\|\bar{p}_{k-1}\|_{\lambda_k}{}^2, \;\; \lambda_k = diag\{\Lambda_k\} \quad (EMSE) \tag{2.53}$$

$$\xi_K = \zeta_k + {\sigma_{v,k}}^2 \quad (MSE) \tag{2.54}$$

The equation eq.2.51 contains information from two spatial locations. The a set of N coupled equations obtained by iterating the equation eq.2.51 are given by

$$E\|\bar{p}_1\|_{\bar{\sigma}_1}{}^2 = E\|\bar{p}_1\|_{\bar{F}_1\bar{\sigma}_1}{}^2 + g_1\bar{\sigma}_1$$

$$E\|\bar{p}_2\|_{\bar{\sigma}_2}{}^2 = E\|\bar{p}_2\|_{\bar{F}_2\bar{\sigma}_2}{}^2 + g_2\bar{\sigma}_2$$

$$\vdots$$

$$E\|\bar{p}_{k-2}\|_{\bar{\sigma}_{k-2}}{}^2 = E\|\bar{p}_{k-3}\|_{\bar{F}_{k-2}\bar{\sigma}_{k-2}}{}^2 + g_{k-2}\bar{\sigma}_{k-2} \tag{2.55}$$

$$E\|\bar{p}_{k-1}\|_{\bar{\sigma}_{k-1}}{}^2 = E\|\bar{p}_{k-2}\|_{\bar{F}_{k-1}\bar{\sigma}_{k-1}}{}^2 + g_{k-1}\bar{\sigma}_{k-1}$$

$$\vdots$$

$$E\|\bar{p}_N\|_{\bar{\sigma}_N}{}^2 = E\|\bar{p}_{N-1}\|_{\bar{F}_N\bar{\sigma}_N}{}^2 + g_N\bar{\sigma}_N \tag{2.56}$$

These equations need to be solved by properly choosing the free parameters.

By choosing $\bar{\sigma}_{k-2} = \bar{F}_{k-1}\bar{\sigma}_{k-1}$ and substituting in eq.2.55 we get

$$E\|\bar{p}_{k-2}\|^2_{\bar{F}_{k-1}\bar{\sigma}_{k-1}} = E\|\bar{p}_{k-3}\|^2_{\bar{F}_{k-2}\bar{F}_{k-1}\bar{\sigma}_{k-1}} + g_{k-2}\bar{F}_{k-1}\bar{\sigma}_{k-1} \tag{2.57}$$

$$E\|\bar{p}_{k-1}\|^2_{\bar{\sigma}_{k-1}} = E\|\bar{p}_{k-3}\|^2_{\bar{F}_{k-2}\bar{F}_{k-1}\bar{\sigma}_{k-1}} + g_{k-2}\bar{F}_{k-1}\bar{\sigma}_{k-1} + g_{k-1}\bar{\sigma}_{k-1} \tag{2.58}$$

By iterating in this manner, we get the equation involving only $\bar{p}_{k-1}$

$$\begin{aligned} E\|\bar{p}_{k-1}\|^2_{\bar{\sigma}_{k-1}} = E\|\bar{p}_{k-1}\|^2_{\bar{F}_{k}\cdots\bar{F}_N\bar{F}_1\cdots\bar{F}_{k-1}\bar{\sigma}_{k-1}} + g_k\bar{F}_{k+1}\cdots\bar{F}_N\bar{F}_1\cdots\bar{F}_{k-1}\bar{\sigma}_{k-1} + \\ g_{k+2}\bar{F}_{k+2}\cdots\bar{F}_N\bar{F}_1\cdots\bar{F}_{k-1}\bar{\sigma}_{k-1} + \cdots + g_{k-2}\bar{F}_{k-1}\bar{\sigma}_{k-1} + g_{k-1}\bar{\sigma}_{k-1} \end{aligned} \tag{2.59}$$

Let us define N matrices as a of product of $\bar{F}$ matrices

$$\Pi_{k,l} \triangleq \bar{F}_{k+l-1}\bar{F}_{k+l}\cdots\bar{F}_N\bar{F}_1\cdots\bar{F}_{k-1}, \quad l = 1,2,\cdots,N \tag{2.60}$$

Where all the subscripts are *mod N*. $\Pi_{k,l}$ is the transition matrix of the weighting vector $\bar{\sigma}_{k-1}$ to arrive node k in a cyclic order through nodes k-1, k-2, N-1,…, k. Rewriting equation eq.2.59 as

$$E\|\bar{p}_{k-1}\|^2_{(I-\Pi_{k,1})\bar{\sigma}_{k-1}} = a_k\bar{\sigma}_{k-1} \tag{2.61}$$

Where the row vector a_k is defined as

$$a_k \triangleq g_k\Pi_{k,2} + g_{k+1}\Pi_{k,3}\cdots + g_{k-2}\Pi_{k,N} + g_{k-1} \tag{2.62}$$

a_k signifies the total mixed effect of transformed local noise and local data characteristics reaching k[th] node from other nodes over the ring topology. By selecting the weight vector $\bar{\sigma}_{k-1}$ as the solution for the linear equation $(I - \Pi_{k,1})\bar{\sigma}_{k-1} = q$, we arrive at the below expression for the MSD

$$\eta_k = a_k(I - \Pi_{k,l})^{-1}q \quad (MSD) \tag{2.63}$$

Similarly for EMSE choose $\bar{\sigma}_{k-1}$ as the solution of $(I - \Pi_{k,1})\bar{\sigma}_{\zeta,k-1} = \lambda_k$

$$\zeta_k = a_k(I - \Pi_{k,l})^{-1}\lambda_k \quad (EMSE) \tag{2.64}$$

$$\xi_k = \zeta_k + {\sigma_{v,k}}^2 \quad (MSE) \tag{2.65}$$

Presence of $\Pi_{k,l}$ and a_k signifies influence of the entire network on every individual node, with some importance given to the local data and noise statistics λ_k and ${\sigma_{v,k}}^2$.

For small step sizes, $\bar{F}_k \approx I - 2\mu_k\Lambda_k$ i.e. $\bar{F}_k$ has become a diagonal matrix, which implies that $\Pi_{k,l} = \Pi = \bar{F}_1\bar{F}_2 \cdots \bar{F}_N$ will also be diagonal and can be approximated as below:

$$\Pi = (I - 2\mu_1\Lambda_1)(I - 2\mu_2\Lambda_2) \cdots\cdots (I - 2\mu_N\Lambda_N) \tag{2.66}$$

$$\approx I - (2\mu_1\Lambda_1 + 2\mu_2\Lambda_2 + \cdots\cdots + 2\mu_N\Lambda_N) \tag{2.67}$$

So that

$$I - \Pi \approx 2\mu_1\Lambda_1 + 2\mu_2\Lambda_2 + \cdots\cdots + 2\mu_N\Lambda_N \tag{2.68}$$

For small step sizes and form eq.2.63 and eq.2.64 we get

$$\eta_k \approx ({\mu_1}^2{\sigma_{v,1}}^2{\lambda_1}^T + \cdots + {\mu_N}^2{\sigma_{v,N}}^2{\lambda_N}^T)(2\mu_1\Lambda_1 + \cdots\cdots + 2\mu_N\Lambda_N)^{-1}\, q \tag{2.69}$$

$$\zeta_k = ({\mu_1}^2{\sigma_{v,1}}^2{\lambda_1}^T + \cdots + {\mu_N}^2{\sigma_{v,N}}^2{\lambda_N}^T)(2\mu_1\Lambda_1 + \cdots\cdots + 2\mu_N\Lambda_N)^{-1}\lambda_k \tag{2.70}$$

For small step sizes, there is an levelling impact on MSD throughout the network, signifying the intermediate averaging in consensus implementations and EMSE goes asymptotically to zero and so MSE converges to the noise level ${\sigma_{v,N}}^2$.

2.7. Simulation Results & Discussion

Computer simulations are provided comparing with the theoretical performance. The whole analysis is based on independent assumptions and simulations are carried out using regressors with shift structure to cope up with realistic situations. The regression vectors are filled up as below:

$$u_{k,i} = col\{u_k(i), u_k(i-1), ..., u_k(i-M+1)\} \tag{2.71}$$

For generating learning curves, 200 independent experiments are performed and averaged. The steady-state performance curves are generated by running the network learning process for 1000 iterations. The quantities of interest, MSD, MSE, and EMSE, are then obtained by averaging the last 500 samples of the corresponding learning curves.

The measurement data ${d_k}^{(i)}$ are achieved by using the data model in eq.2.12 at each node and the desired M×1vector to be estimated is set as w^0=col {1, 1,…,1}/$\sqrt{M}$, where M is the tap size considered as M=10. The quantities of interest are defined as below:

$$\text{EMSE (Excess Mean square error)} = \left|u_{k,i}(\psi_k{}^{(i)} - \bar{w}^0)\right|^2 \tag{2.72}$$

$$\text{MSE (Mean square error)} = \left|d_k(i) - u_{k,i}\psi_{k-1}{}^{(i)}\right|^2 \tag{2.73}$$

$$\text{MSD (Mean square deviation)} = \left|(\psi_k{}^{(i)} - \bar{w}^0)\right|^2 \tag{2.74}$$

A network of 20 nodes i.e. N=20 is considered, with each input regressors vector of size (1×10) and each node collects time correlated data $u_k{}^{(i)}$ generated as

$$u_k(i) = \alpha_k u_k(i-1) + \beta_k Z(i) \quad , i > -\infty \tag{2.75}$$

$$\text{Where } \beta_k = \sqrt{\sigma_{u,k}{}^2(1-\alpha_k{}^2)}$$

Where the correlation index, $\propto_k \in [0,1)$ and $Z(i)$ is a spatially independent WGN random process with zero mean and unit variance. The resulting regressor will have Toeplitz covariance matrix $R_{u,k}$, with correlation sequence $r_k(\text{i}) = \sigma_{u,k}{}^2(\propto_k)^{|i|}$, $i = 0,1,\cdots,M-1$. The input regressor power profile $\sigma_{u,k}{}^2 \in (0,1]$, the correlation index $\propto_k \in (0,1]$ and the Gaussian noise variance $\sigma_{v,k}{}^2 \in (0,0.1]$ are chosen randomly and are depicted in Fig.2.6, Fig.2.7, Fig.2.8.

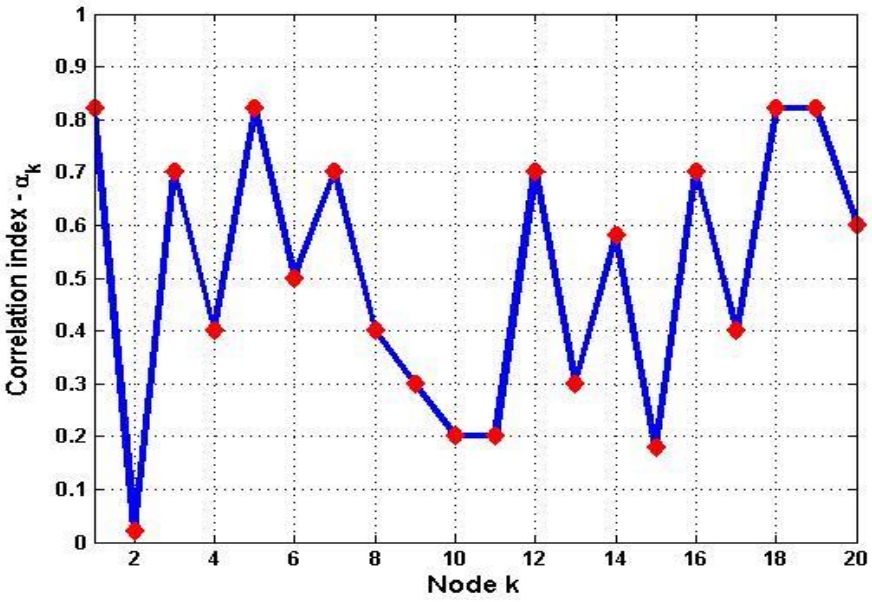

Fig.2.6. Correlation Index at each Node

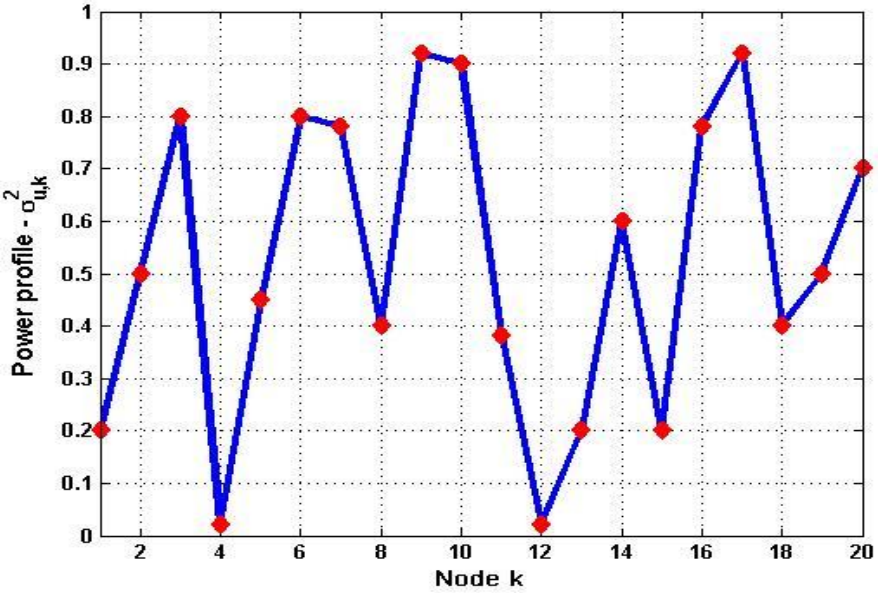

Fig.2.7. Regressor power profile at each Node

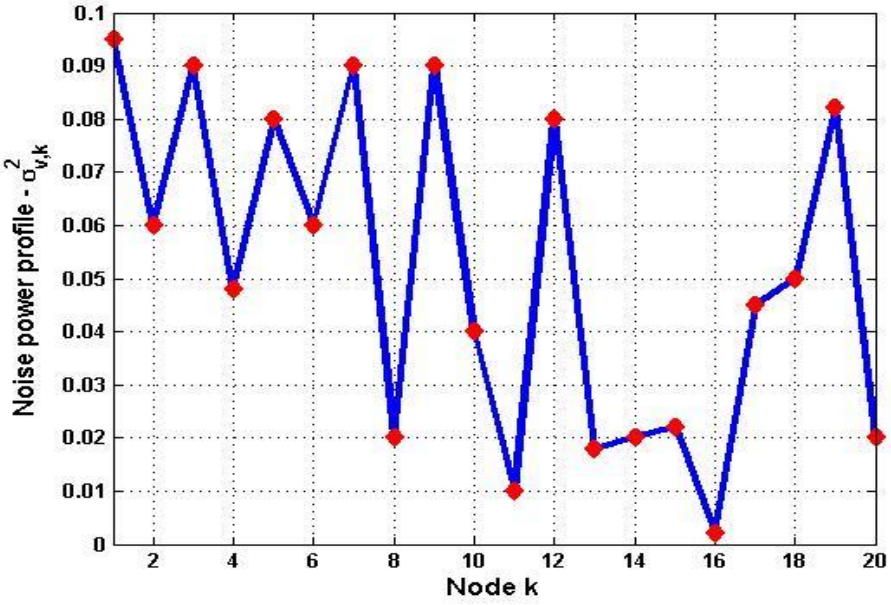

Fig.2.8. Noise power profile at each Node

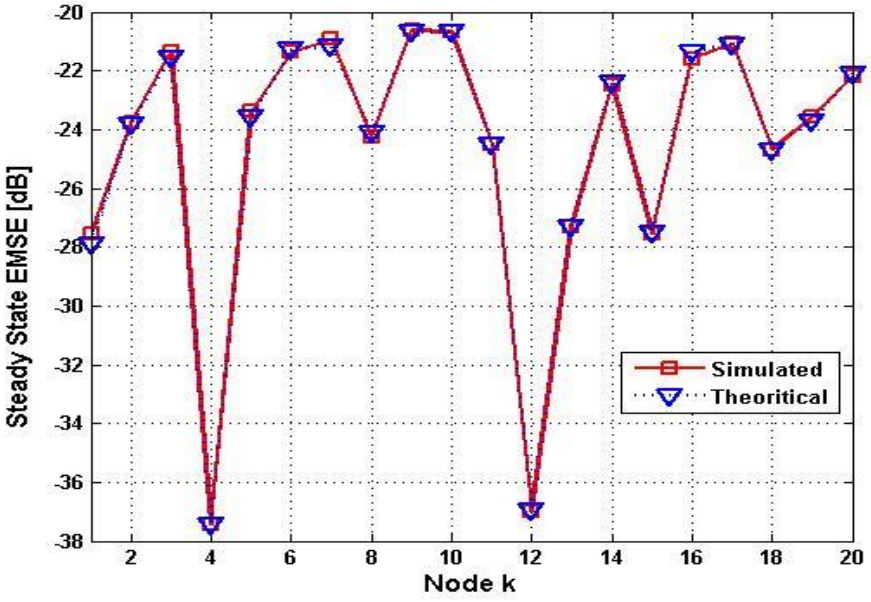

Fig.2.9. Comparison of simulated and theoretical steady state EMSE at each node

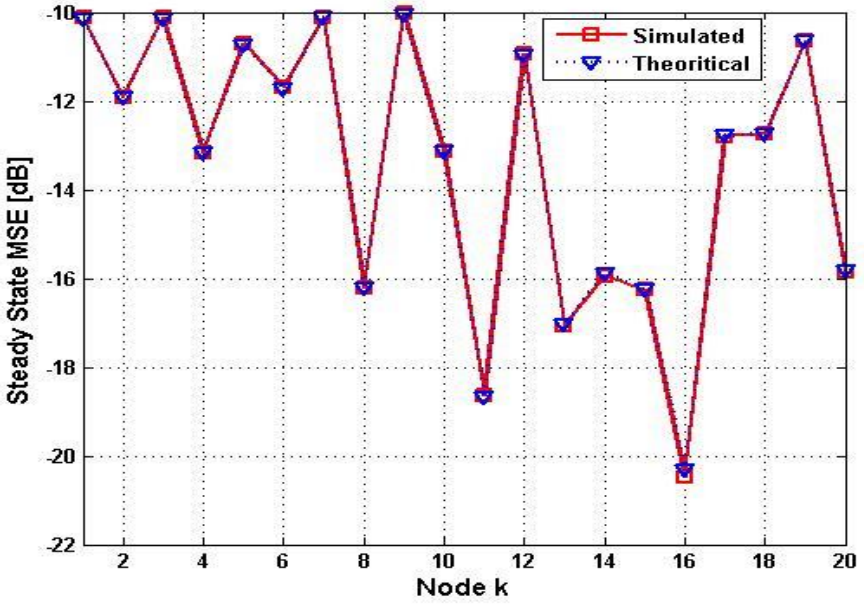

Fig.2.10. Comparison of simulated and theoretical steady state MSE at each node

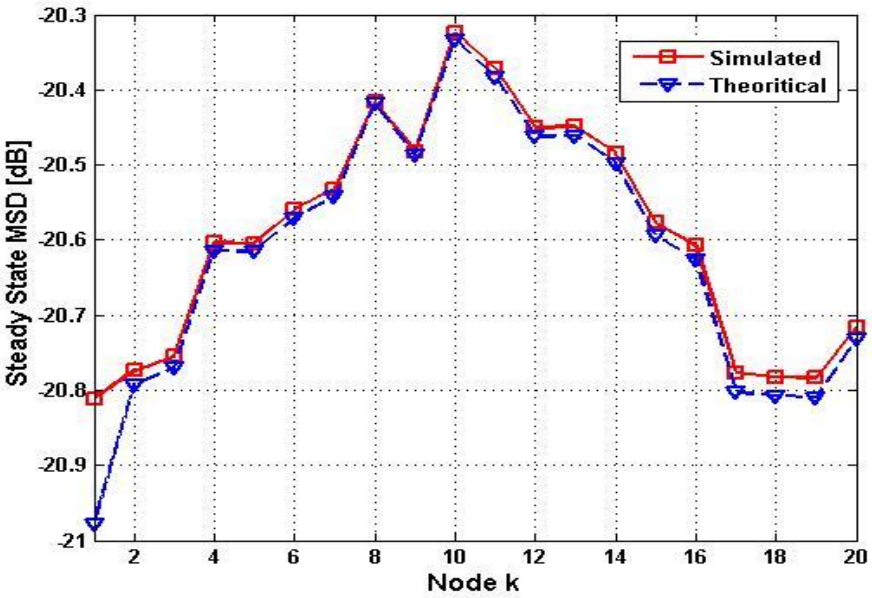

Fig.2.11. Comparison of simulated and theoretical steady state MSD at each node

There is an excellent match between the theoretical and simulated steady state values for significantly larger step sizes. Though the statistical profile of noise and input is different at all the nodes, the Mean square deviation is approximately even throughout the network with a very less deviation from -20.6 db. EMSE and MSE are more vulnerable to the local statistics. MSE reflects the noise power at each node in the distributed network considered.

CHAPTER 3

DISTRIBUTED INCREMENTAL LLMS

3.1. Introduction

The incremental adaptive solution using LMS algorithm suffers from drift problem in non-ideal situations [7] [8]. LMS is widely used due to its ease of implementation, but it suffers from drift problem when implemented in finite precision environment. It is vulnerable to ill conditioning or inadequate excitation of the input [9]. In this chapter we propose Incremental LLMS algorithm to overcome the drift problem of LMS [10] [11]. Leaky LMS is a modified version of LMS where small leakage is allowed in the update equation and it introduces some bias in the parameter estimate. The Eigen spread of Leaky LMS is slightly lesser than LMS algorithm. So leaky LMS converges faster than LMS in case of inputs with high Eigen spread [17].

3.2. Weight Drift problem with LMS

Conventional LMS algorithm is the most generally used due to its simplicity, less computational complexity and ease of implementation. The optimization function or the cost function of LMS criterion is

$$\min_w \left[J(w) \triangleq E\left|d - uw\right|^2 \right] \tag{3.1}$$

Solving the least mean squares criterion results in the weight estimate update equation

$$w(k) = w(k-1) + \mu(k)u(k)e(k) \tag{3.2}$$

Where $w(k)$ is the filter weight vector, $\mu(k)$ is step size, $u(k)$ is the input sequence and

$$e(k) = d(k) - u^T(k)w(k-1)$$

$$d(k) = u^T(k)w_0 + v(k)$$

w_0 refers to the optimum solution.

LMS filter can produce unbounded weight estimates in non-ideal or practical, which is

referred as drift problem. The convergence and stability of LMS algorithm may be problematic in non-ideal conditions. In such cases the weight estimates don't converge to the optimum value and go unbounded, i.e. they diverge. So the incremental adaptive solution using LMS algorithm suffers from drift problem in practical implementations. The difficulties with LMS algorithm are discussed below:

3.2.1. Finite Precision Effects

In digital implementation of the adaptive filters, all the inputs and the intermediate values are quantized using an analog to digital converter as shown in Fig.3.1. This quantization results in some quantization error, which relies on the number of levels utilized and the threshold value of the quantizer. The quantization error gets accumulated continuously until an overflow occurs. The overflow is unacceptable for any application where the system is operated continuously. A real time system processes quantization errors in an unstable manner which doesn't affect the performance immediately; it may take hours or days. The reason for unexpectedly large inaccuracies is the continuous accumulation of quantization errors with time until their effects reach a level that causes the adaptive filter performance to be unacceptable [7] [8].

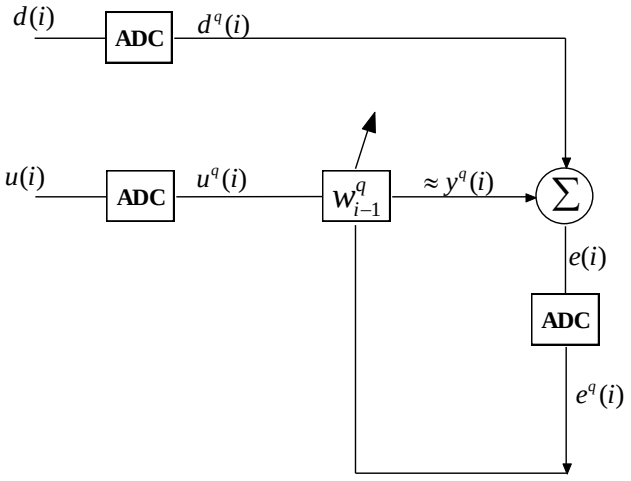

Fig. 3.1. Block diagram representation of finite precision implementation of adaptive filter

All the coefficients and quantities are stored in registers whose word length is finite. So all the values are truncated to some precision in order to store them in registers, which results in finite precision errors. Finite precision implementation results in bias in the estimate from the

ideal infinite precision conditions. These finite precision errors get accumulated continuously with time and it results in overflow. It is not a good idea to increase the precision as the implementation cost is strongly affected by the total number of bits used for the digital implementation. The noise or the finite precision arithmetic errors will become non-zero mean variables due to finite precision errors.

3.2.2. Ill Conditioning of the input

The ill condition of the input means that there is wide Eigen spread, where Eigen spread is the ratio of the largest Eigen value to the least Eigen value of the auto correlation matrix of the input.

$$\text{Eigen Spread} = \frac{\lambda_{\max}}{\lambda_{\min}}$$

where $\lambda_{\max}$ and $\lambda_{\min}$ are the eigen values of auto correlation matrix $R_u = \mathrm{E}\left\{u.u^H\right\}$. It results in near singular auto correlation matrix, which is makes the estimate to slowly escape from the expected value to infinity. Even though all the other signals are finite, the parameter estimate will go unbounded due to inadequacy of excitation [8] [9]. The effect of singular auto correlation matrix can be explained alternatively as given below:

The error function of the LMS can be written as

$$\xi = \xi_{\min} + \left(w - w_0\right)^H R_u \left(w - w_0\right) \tag{3.3}$$

Using Eigen value decomposition for the auto correlation matrix

$$R_u = V \Lambda V^H$$

Substituting in eq 3.3, we get

$$\xi = \xi_{\min} + \left(V^H \left(w - w_0\right)\right)^H \Lambda V^H \left(w - w_0\right) \tag{3.4}$$

If R_u is singular, it corresponds to the existence of the eigen value $\lambda_i = 0$. So the cost function could be $\xi = \xi_{\min}$, even though $V^H \left(w - w_0\right) \neq 0$ i.e. w will not be equal to the optimum weight w_0, which implies that w will run away without effecting ξ i.e. drift problem.

3.2.3. Numerical Stability

An algorithm is said to be numerically stable if it limits the maximum deviation from the infinite precision implementation, whereas a numerically unstable algorithm allows the errors to accumulate with time, which results in divergence of the estimate. Increasing the precision does not affect the numerical stability [7]. Numerical stability is a strong function of the algorithm. A numerically stable algorithm keeps track of the finite precision errors and corrects itself.

3.2.4. Accuracy

Accuracy of an algorithm implementation is the magnitude of the deviation from the infinite precision performance [8]. The smaller deviation indicates more accuracy. Accuracy greatly depends on the number of bits used for storage i.e. precision.

So LMS algorithm suffers from numerical instability, accuracy affects and is sensitive to ill-conditioning of the input sequence. The incremental adaptive solution provided for parameter estimation in distributed networks using LMS algorithm will result in unbounded parameter estimate in practical implementation. Leaky LMS solves the drift problem and this chapter deals with implementation of Incremental Leaky LMS for parameter estimate in distributed networks.

3.3. Proposed Framework

Incremental Leaky LMS [21] is proposed to overcome the drift problem arising in ILMS. Leaky LMS is modified version of the LMS algorithm. The optimization equation for Leaky LMS is

$$\min_w \left[J^{\alpha}(w) \triangleq \alpha \|w\|^2 + E|d - uw|^2 \right] \tag{3.5}$$

Where α - positive real number referred as leakage factor ranging from 0 to 1.

$$0 < \alpha < 1$$

The update equation for the LLMS algorithm [11] [12] is

$$w(k) = (1 - \mu(k)\alpha)w(k-1) + \mu(k)u(k)e(k) \tag{3.6}$$

The subsequent analysis follows the following data model and the assumptions done for the algorithm implementation and performance analysis are listed below:

3.3.1. Data Model and Assumptions

1. The desired unknown vector w_0 relates $\{u_{k,i}, d_k(i)\}$ as
$$d_k(i) = u_{k,i}w_0 + v_k(i) \tag{3.7}$$
2. $v_k(i)$ is white Gaussian noise with variance $\sigma^2_{v,k}$ and independent of $\{u_{k,i}, d_k(i)\}$ for all i,j
3. Input sequence $u_{k,i}$ is spatial and time independent.
4. The input is assumed to be corrupt with white Gaussian noise $n_{k,i}$, with zero mean and $\sigma^2_{n,k}$ variance
$$z_{k,i} = u_{k,i} + n_{k,i} \tag{3.8}$$

Fig.3.2. shows the data flow and the weight updation in Incremental Leaky LMS strategy in a distributed network.

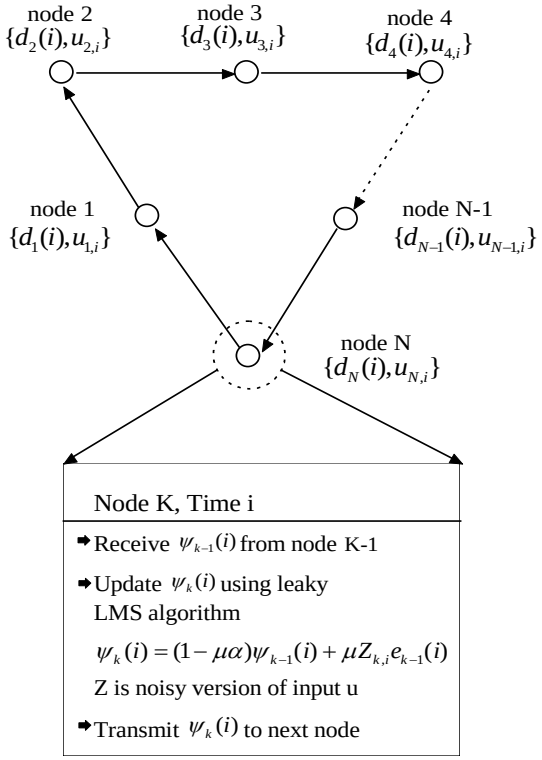

Fig. 3.2. Data Flow and updation in Incremental leaky LMS

Algorithm for Incremental Leaky LMS Solution

Let $\psi_k^{(i)}$ be the local estimate of w_0 at node k at time i.

Start with $w_{-1} = 0$

For each time $i \geq 0$, repeat :

Set $\psi_0^{(i)} = w_{i-1}$

For nodes $k = 1$ to N, repeat :

Receive $\psi_{k-1}^{(i)}$ from previous node

$$\psi_k^{(i)} = (1 - \mu_k \alpha)\psi_{k-1}^{(i)} + \mu_k z_{k,i}^{*}(d_k(i) - u_{k,i}\psi_{k-1}^{(i)}) \quad k = 1, 2, ..., N$$

End

$w_i = \psi_N^{(i)}$

Send w_i to node 1

End

3.3.2. Performance Analysis

In this section an example is given to illustrate the weight drift problem occurring in LMS update equation and how leaky LMS solves the problem [16]. Let's consider that the input vector $u(k)$ is orthogonal to weight error vector

$$w(k) = w_0 - w(k-1) \tag{3.9}$$

Now the LMS update equation is

$$w(k) = w(k-1) + \mu(k)u(k)e(k)$$

Where $e(k) = d(k) - u^T(k)w(k-1)$

$$d(k) = u^T(k)w_0 + v(k)$$

The weight error vector satisfies the equation

$$w(k) = w(k-1) - \mu(k)u(k)v(k) \tag{3.10}$$

Taking norm on both sides the above eq.3.10 can be written as

$$\left\|w(k)\right\|^2 = \left\|w(k-1)\right\|^2 + \mu^2(k)\left\|u(k)\right\|^2 v^2(k)$$

Solving this recursion for $w(N)$, we get

$$\left\|w(N)\right\|^2 = \sum_{K=1}^{N} \mu^2(k)\left\|u(k)\right\|^2 v^2(k) + \left\|w(0)\right\|^2 \tag{3.11}$$

From above eq.3.11, it is obvious that $\left\|w(N)\right\|^2 \to \infty$ with N if $\mu(k)\,||\,u(k)\,||\,v(k)$ is a power sequence or not a finite energy sequence. This situation doesn't happen when LLMS is utilized, as weight update equation contains leakage factor.

LLMS weight update equation is

$$w(k) = (1-\alpha)w(k-1) + \mu(k)u(k)e(k) \tag{3.12}$$

Taking norm on both sides the above eq.3.12 results as

$$\left\|w(k)\right\|^2 = (1-\alpha)\left\|w(k-1)\right\|^2 + \mu^2(k)\left\|u(k)\right\|^2 v^2(k) \tag{3.13}$$

From eq.3.13, $\left\|w(k)\right\|^2$ remains bounded for $o\langle\alpha\langle 1$

3.3.2.1. Convergence in the Mean

The Leaky LMS weight update equation is

$$w(k) = (1-\alpha)w(k-1) + \mu(k)u(k)e(k) \tag{3.14}$$

The weight error vector satisfies the equation

$$w(k) = (1-\mu\alpha)w(k-1) + \mu\alpha w_0 - \mu u(k)e(k) \tag{3.15}$$

Taking expectation on both sides the above eq.3.15

$$E\left[w(k)\right] = (1-\mu\alpha)E\left[w(k-1)\right] + \mu\alpha w_0 - \mu E\left[u(n)e(n)\right]$$

$$E\left[w(k)\right] = \left[1 - \mu\left[\alpha I + R_u\right]\right] E\left[w(k-1)\right] + \mu\alpha w_0 \tag{3.16}$$

The mean $E\left[w(k)\right]$ converges to zero, and consequently $E\left[w(k)\right]$ converges to w_0 if and only if $0 \langle \mu \langle \frac{2}{\alpha + \lambda_{\max}}$, where $\lambda_{\max}$ is the largest Eigen value of the matrix $R_u = E[u(n)u(n)^T]$. In other words Leaky LMS is convergent in mean, if the stability condition is met [12].

3.3.2.2. Solution to ill-conditioned input (Wide Eigen Spread)

For leaky LMS with leakage factor α, the optimum weight is $w_{opt} = \left[R_u + \alpha I\right]^{-1} R_{du}$, where R_u is the auto correlation of the input sequence and R_{du} is the cross correlation between input and the desired data.

Let λ_{max}, λ_{min} be the maximum and minimum Eigen values of autocorrelation matrix, then Eigen spread for LMS algorithm is $\frac{\lambda_{\max}}{\lambda_{\min}}$

Whereas Eigen spread for Leaky LMS will be $\frac{\lambda_{\max} + \alpha}{\lambda_{\min} + \alpha}$, as the new auto correlation matrix is $R_u + \alpha I$.

As $\alpha \geq 0$, the Eigen spread in case of LLMS is less than that of LMS [17]

$$\frac{\lambda_{\max} + \alpha}{\lambda_{\min} + \alpha} \leq \frac{\lambda_{\max}}{\lambda_{\min}} \tag{3.17}$$

So Leaky LMS overcomes the sensitivity to ill-conditioned input sequence.

3.4. Discussions

An efficient approach is proposed to tackle the drift problem that arises in distributed incremental LMS approach due to finite precision effects, quantization errors, inadequate or ill-conditioned inputs by implementing Leaky LMS algorithm for distributed processing using incremental strategy. Incremental Leaky LMS solves the drift problem, by introducing leakage factor which results in the energy leakage preventing weight to go unbounded. But it introduces bias in the mean value of parameter estimate i.e. it will not converge to optimum value.

CHAPTER 4

INCREMENTAL MODIFIED LLMS

4.1. Introduction

LLMS is proposed to overcome the problem of parameter drift arising in conventional LMS algorithm [10] [16]. Although LLMS mitigates the drift issue, the overall performance is same as that of LMS algorithm. Modified Leaky LMS is the enhanced version, which overcomes drift problem as well as provides better performance than LLMS [19]. This better performance is accomplished at the cost of slightest increase in the computational complexity.

4.2. Modified LLMS

The LMS algorithm is one of the most famous adaptive algorithms for linear estimation due to its simplicity and ease of implementation. This has led to the development of variations of LMS algorithm, which are available in the literature. Some of the improved versions of LMS include NLMS, sign LMS, variable step size LMS, sign error LMS, sign regressor LMS etc... All these improved versions are developed to achieve faster convergence and better MSE.

One of the main difficulties facing with LMS algorithm is the drift problem, where the parameter estimate will diverge despite of the bounded input conditions [7] [8]. The leaky LMS (LLMS) is a modified version of conventional LMS algorithm and it overcomes the drift problem by bounding the parameter estimate using the leakage factor in the weight update equation. LLMS also solves the problems like stalling and improves stability, tracking capability. The main drawback of LLMs is its convergence speed. Though LLMS solves the drift problem, the convergence speed and MSE performance is almost same as that of LMS algorithm. A novel algorithm is proposed in the literature to improve the convergence speed based on the Least Sum of Exponentials algorithm (LSE) [20].

LMS uses the second order error function, such Second Order Statistic (SOS) algorithms are very easy to implement and have less computational complexity [10]. Higher order error power algorithms like Least Mean Fourth (LMF) algorithm comes under Higher Order

Statistic (HOS) algorithms, which have high computational complexity, faster convergence rate and instability issues [10] [18]. To make use of both the advantages of SOS and HOS, mixed norm gradient descent algorithms have been developed [20]. LSE is one of such mixed norm gradient descent algorithm, which considers infinite number of error powers in the cost function. LSE algorithm employs sum of exponentials of errors in the cost function, which is the generalization of the mixed norm stochastic gradient algorithms. The cost function of the Modified Leaky LMS (MLLMS) algorithm is defines as below:

$$J(k) = \left(\exp\left(e(k)\right) + \exp\left(-e(k)\right)\right)^2 + \gamma w^T(k) w(k) \tag{4.1}$$

So the error surface of the cost function defined in eq.4.1 is smooth and convex, which improves the convergence speed. MLLMS is the modification of LSE [19]. The error surface will be steeper and so the convergence speed is faster than the LLMS algorithm [20].

Where $e(k)$ is the error defined as below:

$$e(k) = d(k) - u^T(k) w(k-1) \tag{4.2}$$

Differentiating eq. 4.1 with respect to $w(k)$, we get

$$\frac{\delta J(k)}{\delta w(k)} = 2\left(-u(k)\exp\left(e(k)\right) + u(k)\exp\left(-e(k)\right)\right) + 2\gamma w(k) \tag{4.3}$$

Now the weight update equation is given as:

$$w(k) = w(k-1) - \frac{\mu}{2}\frac{\delta J(k)}{\delta w(k)} \tag{4.4}$$

Substituting eq.4.3 in eq.4.4, the resulting update equation is as below:

$$w(k+1) = (1 - \mu\gamma) w(k) + 2\mu u(k) \sinh\left(e(k)\right) \tag{4.5}$$

4.3. Proposed Framework

Incremental Modified Leaky LMS algorithm is proposed to improve the performance of ILLMS. So IMLLMS will overcome drift problem with improved performance compared to ILMS.

4.3.1 Data Model and Assumptions

The data model and the assumptions used for carrying out the performance analysis of the adaptive algorithm are listed below:

- The desired unknown vector w^0 relates $\{d_k(i), u_{k,i}\}$ as

$$d_k(i) = u_{k,i}w^0 + v_k(i) \tag{4.6}$$

  Where $v_k(i)$ is white noise sequence with variance $\sigma_{v,k}{}^2$ and independent of $\{d_l(j), u_{l,j}\}$ for all l,j.

- $u_{k,i}$ is independent of $u_{l,i}$ for k≠ l (Spatial independence).
- $u_{k,i}$ is independent of $u_{k,j}$ for $i \neq j$ (time independence).

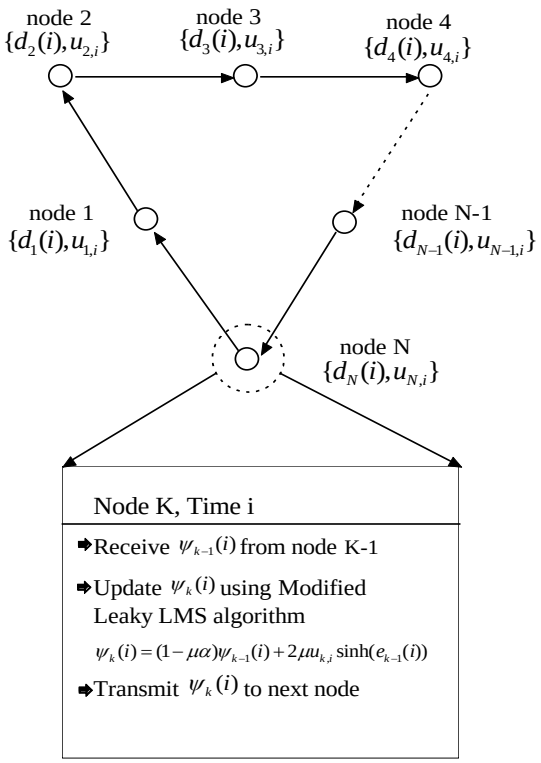

Fig. 4.1. Data Flow and updation in Incremental modified leaky LMS

Fig.4.1. shows the data flow and the weight updation in IMLLMS strategy in a distributed adaptive sensor network with N nodes.Incremental Modified LLMS algorithm can be shown as below:

Algorithm for Incremental Modified LLMS Solution

Let $\psi_k^{(i)}$ denote a local estimate of w_0 at node k at time i.

Start with $w_{-1} = 0$

For each time $i \geq 0$, repeat :

Set $\psi_0^{(i)} = w_{i-1}$

For nodes $k = 1$ to N, repeat :

Receive $\psi_{k-1}^{(i)}$ from previous node

$$\psi_k^{(i)} = (1 - \mu_k \alpha)\psi_{k-1}^{(i)} + 2\mu_k u_{k,i}^* \sinh(d_k(i) - u_{k,i}\psi_{k-1}^{(i)}) \quad k = 1, 2, ..., N$$

End

$w_i = \psi_N^{(i)}$

Send w_i to node 1

End

4.4. Simulation Results & Discussion

The computer simulations are provided by performing 300 independent experiments and averaging. ILLMS and IMLLMS are implemented and then compared in terms of convergence speed, MSE, MSD and EMSE. The input at each node is considered as shift structure in order to cope up with the realistic scenarios. The regression vectors are filled up as below:

$$u_{k,i} = col\{u_k(i), u_k(i-1), ..., u_k(i-M+1)\}$$

The measurement data $d_k^{(i)}$ are generated at each node by using the regular data model as mentioned in eq.4.6 and the desired M×1vector to be estimated is set as w^0=col {1, 1,…,1}/$\sqrt{M}$, where M is the tap size and take as M=5. The quantities of interest are defined as below:

EMSE (Excess Mean square error) = $\left|u_{k,i}(\psi_k{}^{(i)} - \bar{w}^0)\right|^2$ (4.7)

MSE (Mean square error) = $\left|d_k(i) - u_{k,i}\psi_{k-1}{}^{(i)}\right|^2$ (4.8)

MSD (Mean square deviation) = $\left|(\psi_k{}^{(i)} - \bar{w}^0)\right|^2$ (4.9)

A network of 20 nodes is considered in this experiment i.e. N=20 with each input regressor of size (1×5). The input is created by a first order auto regressive model given as below:

$$u_k(i) = 0.2u_k(i-1) + \eta_{0k}(i) \tag{4.6}$$

Where $\eta_0(k)$ is a WGN with mean zero and variance $\sigma^2_{n_0}$ =0.36. The input signal is assumed to be corrupt with white Gaussian noise with zero mean and variance $\sigma^2_{v_0}$ =0.0001. Step size taken as $\mu = 0.003$ and leakage factor considered as $\alpha = 0.01$. The learning curves for MSE, EMSE, MSD for IMLLMS and ILLMS at node 1 are shown in Fig.4.2, Fig.4.3, Fig 4.4.

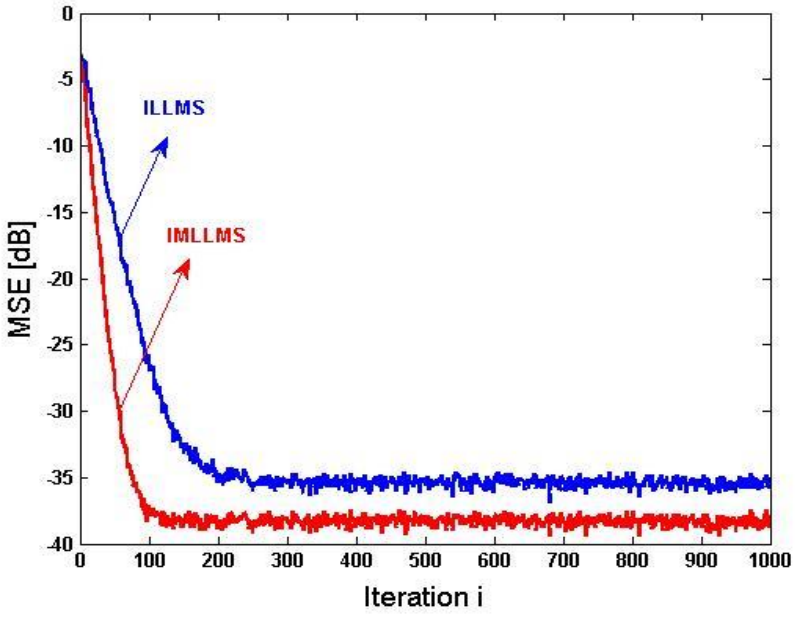

Fig.4.2. MSE at node 1 for Incremenatl LLMS and Incremental MLLMS

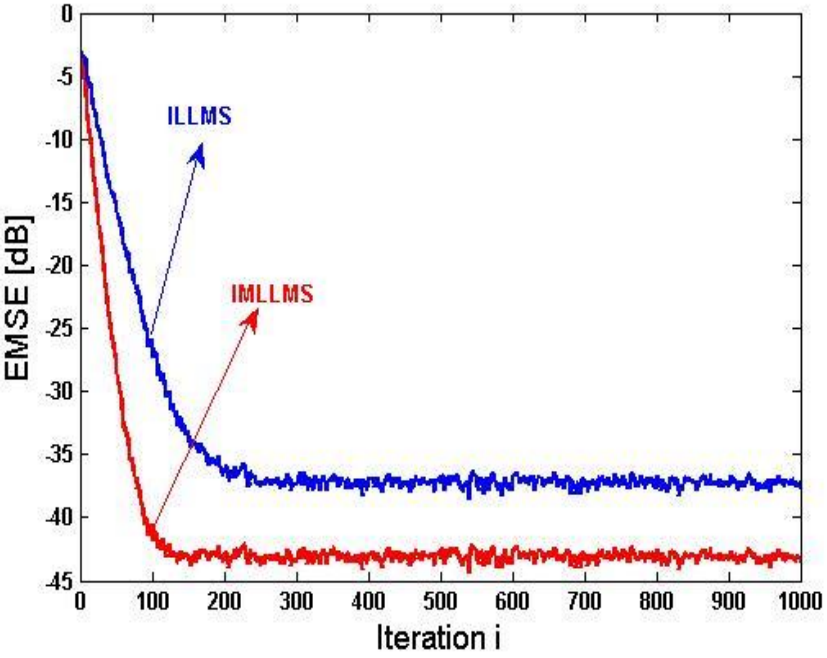

Fig.4.3. EMSE at node 1 for Incremenatl LLMS and Incremental MLLMS

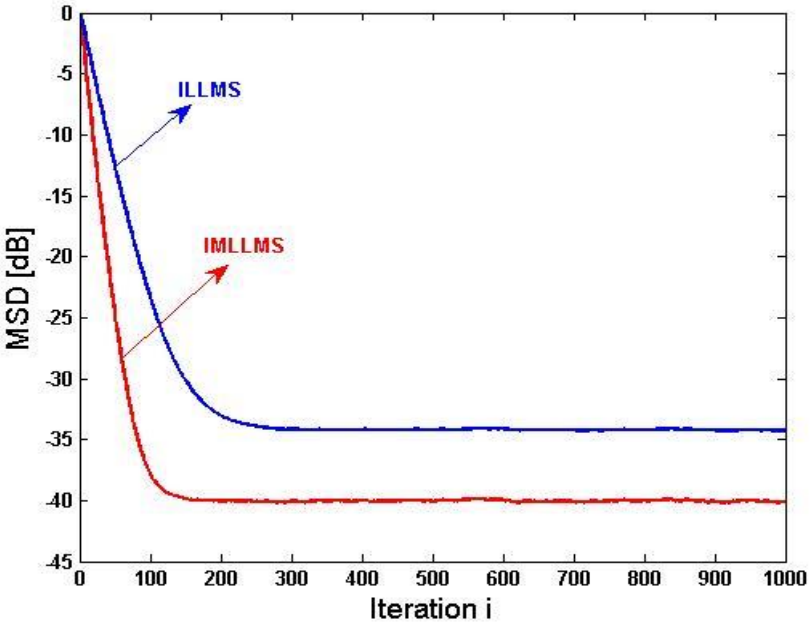

Fig.4.4. MSD at node 1 for Incremenatl LLMS and Incremental MLLMS

A new algorithm has been proposed which improves the performance of ILLMS by implementing MLLMS in place of LLMS, which is obtained by slight modification of cost function of LLMS according to the LSE algorithm. Simulation results show that the IMLLMS algorithm outperforms the ILLMS in terms of convergence rate and the steady state performance in the presence of white Gaussian noise.

CHAPTER 5

CONCLUSION AND FUTURE WORK

Distributed sensor signal processing has a huge scope due to its wide range of applications. The parameter of interest has to be estimated using a variety of algorithms that have been developed for this purpose. The centralized solution for the parameter estimation in wireless sensor networks requires a high power central processor and huge amounts of communication between the nodes. An incremental adaptive distributed solution minimizes the communication burden and power consumption. In this thesis work various adaptive algorithms like LMS, LLMS, and MLLS are implemented for parameter estimation. The choice of the adaptive algorithm depends on the requirement, computational complexity and the convergence rate. Incremental LMS is the simplest algorithm used for parameter estimation in a distributed network containing independent nodes which are adaptive. Simulation results of ILMS conclude that the MSD is uniform at all the nodes in the network, even though all the nodes are having different noise levels, which signifies good performance of the network. MSE reflected the background noise and there is an excellent match between the simulation and theory results.

An efficient approach to solve the drift problem that arises in distributed incremental LMS approach due to finite precision effects, quantization errors, inadequate or ill-conditioned inputs has been developed by implementing Leaky LMS algorithm for distributed processing using incremental strategy. Incremental Leaky LMS solves the drift problem, by introducing leakage factor which results in the energy leakage preventing weight to go unbounded. But it introduces bias in the mean value of parameter estimate i.e. it will not converge to optimum value. Mathematical analysis is provided to explain the drift problem for LMS and to show how LLMS solves it. To improve the performance of ILMS and ILLMS, IMLLMS is proposed. IMLLMS algorithm converges faster with a better steady state performance with a slight increase in the computational complexity. IMLLMS have both the benefits of improved performance and drift problem removal.

All the adaptive algorithms are applied under Gaussian noisy environment. The work can be extended to impulsive noise, where outliers come into picture. IMLLMS can be used

to deal impulse noise and Gaussian correlated noise. Incremental mode of cooperation is considered throughout the work, but malfunctioning of a single node in this mode results in network failure. All the algorithms implemented in this thesis can be implemented in other modes of cooperation which could be the future work. Block Leaky LMS and Block Modified Leaky LMS can be implemented in incremental mode so that the computational complexity decreases.

PUBLICATION

- ✓ M. Sowjanya, A.K. sahoo and Sananda Kumar, "Distributed Incremental Leaky LMS", In proc. of 4th IEEE *International Conference on Communication and Signal Processing*, pp. 1768-1772, April 2-4, 2015.

REFERENCES

[1] D. Estrin, G. Pottie, and M. Srivastava, "Intrumenting the world with wireless sensor setworks," in Proc. *IEEE Int. Conf. Acoustics, Speech, Signal Processing (ICASSP),* pp. 2033–2036,May 2001.

[2] I. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "A survey on sensor networks," *IEEE Commun. Mag.*, vol. 40, no. 8, pp. 102–114, Aug. 2002.

[3] L. A. Rossi, B. Krishnamachari, and C.C. J. Kuo, "Distributed parameter estimation for monitoring diffusion phenomena using physical models," in *Proc. 1st IEEE Conf. Sensor Ad Hoc Communications Networks*, pp. 460–469, 2004.

[4] C. G. Lopes, A. H. Sayed , "Incremental Adaptive Strategies Over Distributed Networks", *IEEE Transaction on Signal Processing,* Vol. 55, No. 8, 2007.

[5] A. Khalili, M. A. Tinati and Amir R. , "An incremental block LMS algorithm for distributed adaptive estimation", *IEEE International Conference on Communication Systems (ICCS),* pp.493 – 496, 2010.

[6] C. G. Lopes and A. H. Sayed, "Distributed processing over adaptive networks," in *Proc. Adaptive Sensor Array Processing Workshop*, MIT Lincoln Lab., Lexington, MA, Jun. 2006.

[7] J. M.Cioffi, "Limited-Precision Effects in Adaptive Filtering", *IEEE Transaction on Circuits and Systems,* Vol. CAS-34, No. 7, 1987.

[8] Sethares, Lawrence, Johnson C.R, Bitmead R.R., "Parameter Drift in LMS Adaptive Filters", *IEEE Transaction on Acoustics, Speech and Signal Processing,* Vol. 34, pp. 868-879, 1986.

[9] A. H. Sayed, *Fundamentals of Adaptive Filtering*. New York: Wiley, 2003.

[10] R Abdolee, B Champagne, A. H. Sayed, "A Diffusion LMS Strategy For Parameter Estimation in Noisy Regressor Applications", *20th European Signal Processing Conference (EUSIPCO 2012),* pp. 749-753, 2012.

[11] A. A. Zerai, and J. A. Bucklew, "Failure Time Analysis for LMS Algorithms", IEEE *Transaction on Information Theory,* Vol. 48, No. 4, 2002.

[12] K. Mayyas and T. Aboulnasr, **"**Leaky LMS Algorithm: MSE Analysis for Gaussian Data" IEEE *IEEE Transaction on Signal Processing,* Vol. 45, No. 4, 1997.

[13] S. C. Douglas, **"**Performance Comparison of Two Implementations of the Leaky LMS Adaptive Filter"**,** *IEEE Transaction on Signal Processing,* vol. 45, No. 8, 1997.

[14] B. D. Rigling and P. Schniter,, "Subspace Leaky LMS", *IEEE Signal Processing Letters*, Vol. 11, No. 2, 2004.

[15] *V. H. Nascimento and A. H. Sayed, "*Unbiased and stable leaky-based adaptive filters*," IEEE Transactions on Signal Processing,* vol. 47, No.12, pp. 3261-3276, 1999.

[16] Nascimento, Sayed A.H., **"**An unbiased and cost-effective leaky-LMS filter**",** *Conference Record of the Thirtieth Asilomar Conference on Signals, Systems and Computers*, Vol. 2, pp. 1078-1082, 1996

[17] M. Kamenetsky and B. Widrow." A Variable Leaky LMS Adaptive Algorithm", *Conference Record of the Thirty-Eighth Asilomar Conference on Signals, Systems and Computers*, Vol. 1, pp. 125-128, 2004.

[18] E. Walach and B. Widrow, "The least mean fourth (LMF) adaptive algorithm and its family", *IEEE Transactions on Information Theory*, vol. 30, No. 2, pp. 275-283, 1984.

[19] T. R. Gwadabe, M. S. Salman, and H. Abuhilal, "A Modified Leaky-LMS Algorithm" *International Journal of Computer and Electrical Engineering,* Vol. 6, No. 3, 2014.

[20] C. Boukis, D. P. Mandic, and A. G. Constantinides, "A generalisemixed norm stochastic gradient algorithm," *in Proc. 15th InternationaConference on Digital Signal Processing*, pp. 27-30 , 2007.

[21] M. Sowjanya, A.K. sahoo and Sananda Kumar, "Distributed Incremental Leaky LMS", In proc. of 4th IEEE *International Conference on Communication and Signal Processing*, pp. 1768-1772, April 2-4, 2015.